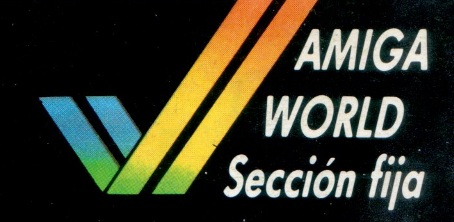




N.º 60 - JUNIO 1989 - Publicación de IDG Communications - 400 Ptas.

ESPEJO

RUTINA
CODIGO
MAQUINA

SUPER SPRITE

SECCION DE JUEGOS

- AFRICAN RAIDERS
- SUPERMAN
- MUNCHER



ISABEL TAPIAS

Super Discos Aplicaciones IV

UNA VEZ MAS HEMOS PREPARADO UN CONJUNTO DE PROGRAMAS ESPECIALMENTE PENSADOS PARA LOS USUARIOS DE COMMODORE 64 Y 128



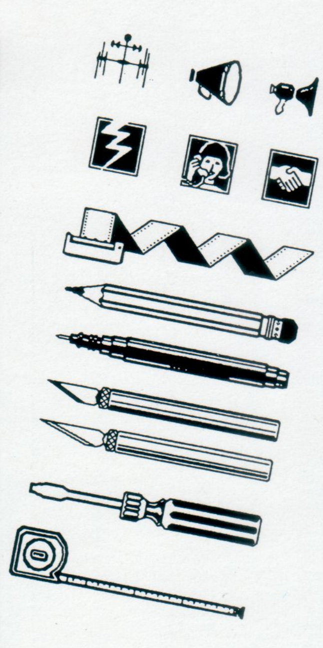
En estos dos discos APLICACIONES IV se encuentran reunidos programas totalmente nuevos (no publicados anteriormente en Commodore World o en otros discos) y algunos programas escogidos de entre los mejores publicados en la revista. Desde que se lanzó el número III, han aparecido algunas utilidades interesantes que hemos considerado oportuno recopilar para todos los usuarios commodorianos.

PROGRAMAS PARA C-64

- **GRAPHMAKER.** Producir e imprimir gráficos tridimensionales con gran sencillez en su C-64 o C-128, en 40 ú 80 columnas.
- **LABEL BASE.** Un completo editor de etiquetas para sus tareas de correos. Podrá crear etiquetas de sus direcciones de bases de datos.
- **RUN PAINT.** El programa artístico que no puede faltar en cualquier programoteca de un commodoriano. Es completísimo y muy bueno, mejor que algunos programas comerciales.
- **RUN TERM.** Por fin podemos presentar un paquete de comunicaciones perfectamente probado en C-64 y C-128. Es sencillo de manejo y superpotente.
- **MAQUINA DE SONIDO 5.0.** La última versión del programa de música más perfeccionado en Commodore World.

(A partir del siguiente, programas publicados en Commodore World)

- **COMPACTOR 64.** Un programa de utilidad que facilita las tareas de los programadores en Basic. Haciendo honor a su nombre, es muy compacto.
- **HELP SYSTEM.** Programa residente en memoria. Sistema de ayuda con calculadora, reloj y otras utilidades.
- **PAGE LISTER.** Listador página a página para permitir comprobar o estudiar mejor los listados de tu C-64.
- **PIANOLA 64.** Un programa de música que, siendo sencillo, posibilita la edición de música por interrupciones.
- **SIZZLE.** Este turbo de disco sirve para acelerar la carga y grabación de programas en modo C-64.
- **SUPERFICIES 3D.** Una maravilla de la generación gráfica tridimensional. Además de útil, se puede aprender sobre programación de gráficos.



Se puede realizar hoy mismo el pedido de estos discos mediante el envío del boletín a:

PROGRAMAS PARA C-128

- **EDITOR MUSICAL 128.** Como su nombre indica se trata de un programa especialmente diseñado aprovechando las características del C-128 para la edición musical.
- Versiones especiales de RUN PAINT y RUN TERM. Para modo 128 también GRAPH MAKER y LABEL BASE.

Además, todos los programas para C-64 funcionan perfectamente en el modo 64 del Commodore 128.

LOS DOS DISCOS AL INCREIBLE PRECIO DE 1.990 ptas.

BOLETIN DE PEDIDO – Especial Aplicaciones IV

Nombre y Apellidos ....

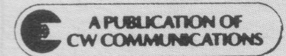
Dirección ....

Población .... C.P. .... Provincia .... Teléfono ....

Forma de Pago: Sólo cheque o giro. No se sirven pedidos contrarreembolso. Gastos de envío e IVA incluidos.

☐ Incluyo cheque por 1.990 ptas.

☐ Envío giro número .... por 1.990 ptas.



Director General:
Francisco Zabala



**Commodore World
está publicado por
CW COMMUNICATIONS, S.A.
y la colaboración
de todos nuestros lectores.**

Director:
Juan Manuel Urraca

Dpto. publicidad:
**Gloria Montalvo (Madrid)
Magda Zabala (Barcelona)**

Redacción Dpto. Técnico:
**Alvaro Ibáñez
Aitor Urraca**

Diseño:
Miguel Angel Hermosell

Distribución y Suscripciones:
**Fernando Rodríguez (dirección),
Angel Rodríguez,
Juan Márquez (suscripciones)
Tel.: 419 40 14**

**COMMODORE WORLD
c/ Rafael Calvo, 18-4.º B
28010 Madrid
Tel. (91) 419 40 14
Télex: 45522
(indicar CW COMMUNICATIONS)
Fax: 419 61 04**

**DELEGACION EN BARCELONA:
c/ Bertrán, 18-20, 3.º - 4.ª
08023 Barcelona
Tels. (93) 212 73 45/212 88 48**

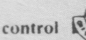
FAX (93) 418 93 55

C.I.F. A.-28-735389

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo, es de 400 ptas., sin I.V.A.

**Distribuidora: SGEL
Avda. Valdelaparra, s/n.
Polg. Ind. de Alcobendas - Madrid**

**Commodore World
es una publicación
IDG COMMUNICATIONS**


control


**Imprime: OMNIA I.G.
Mantuano, 27 - 28002 Madrid
Fotocomposición:
ANDUEZA, S. A.
Fotomecánica: RODACOLOR**

Depósito Legal: M-2944-1984





COMMODORE WORLD es una publicación de IDG Communications, el mayor grupo editorial del mundo en el ámbito informático. IDG Communications edita más de 90 publicaciones relacionadas con la informática en más de 34 países. Catorce millones de personas leen una o más de estas publicaciones cada mes. Las publicaciones del grupo contribuyen al Servicio Internacional de Noticias de IDG que ofrece las últimas noticias nacionales e internacionales sobre el mundo de la informática. Las publicaciones de IDG Communications incluyen: **ARABIA SAUDI:** Arabian Computer News. **ARGENTINA:** Computerworld Argentina. **ASIA:** Communications World; Computerworld Hong Kong; Computerworld South East Asia; PC Review. **AUSTRALIA:** Computerworld Australia; Communications World; Australian PC World; Australian Macworld. **AUSTRIA:** Computerwelt Oesterreich. **BRASIL:** DataNews; PC Mundo; Micro Mundo. **CANADA:** Computer Data. **CHILE:** Informática; Computacion Personal. **COREA DEL SUR:** Computerworld Korea; PC World Korea. **DINAMARCA:** Computerworld Danmark; PC World Danmark; CAD/CAM World. **ESPAÑA:** Computerworld España; PC World España; Commodore World; Comunicaciones World; CIM World. **ESTADOS UNIDOS:** Amiga World; CD-ROM Review; CIO; Computer Currents; Computerworld; Digital News; Federal Computer Week; 80 Micro; FOCUS Publications; InCider; Infoworld; Macintosh Today; Macworld; Computer + Software News (Micro Marketworld/Lebhar-Friedman); Network World; PC Letter; PC World; Portable Computer Review; Publish!; PC Resource; Run. **FINLANDIA:** Mikro; Tietiviikko. **FRANCIA:** Le Monde Informatique; Distributique; InfoPC; Telecoms International. **GRECIA:** Computer Age. **HOLANDA:** Computerworld Netherlands; PC World Benelux. **HUNGRIA:** Computerworld SZT; PC Microvilag. **INDIA:** Dataquest; PC World India. **ISRAEL:** People & Computers Weekly; SBM Monthly. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan; Semicon News. **MEXICO:** Computerworld Mexico; PC Journal. **NORUEGA:** Computerworld Norge; PC World Norge. **NUEVA ZELANDA:** Computerworld New Zealand. **REINO UNIDO:** Computer News; ICL Today; LOTUS; PC Business World. **REPUBLICA FEDERAL ALEMANA:** Computerwoche; Information Management; PC Welt; PCWoche; Run/Run Specials. **REPUBLICA POPULAR CHINA:** China Computerworld; China Computerworld Monthly. **SUECIA:** Computer Sweden; Mirkro Datorn; Svenska PC World. **SUIZA:** Computerworld Schweiz. **VENEZUELA:** Computerworld Venezuela.

EDITORIAL

El código máquina está presente en este número como tema principal de la mayoría de los artículos y programas. Más que una consecuencia de planificación editorial, es el resultado del perfeccionamiento con que los programadores commodorianos elaboran sus rutinas y programas. Nos referimos a los usuarios lectores de la revista y a los colaboradores. Por coincidencia, estos colaboradores han desarrollado para este número rutinas basadas especialmente en código máquina.

La primera de estas rutinas es una parte de un programa más complejo y está dedicada a la inversión de sprites en pantalla. Es muy útil para mover personajes en un juego o para cambiar de dirección un puntero en alguna aplicación en la que se incluyan sprites. Es un primer capítulo que tendrá continuación hasta completar una demostración jugable, de un conocido juego comercial. El juego no saldrá al mercado en versión Commodore debido a su baja calidad y desacierto en el tema, sin embargo nos servirá desde las páginas de esta revista para estudiar los movimientos de los sprites, ahorro de memoria y otros temas de programación.

El Informat, feria de Barcelona que cada año va tomando más relevancia en el mundo informático nacional, abrirá sus puertas el cinco de junio. Nuestra compañía, CW Communications, estará presente en el palacio 2, nivel 1, stand 160. Allí se podrá encontrar tanto Commodore World con sus discos como Amiga World con los correspondientes discos. Una de las probables novedades de la feria será la presentación del A590, módulo de expansión con disco duro para el Amiga 500.

NOTICIAS

AMIGA WORLD 3 ESPECIAL VIDEO

El próximo número de Amiga World estará dedicado al vídeo y la edición de imágenes. En ese número se tratarán temas prácticos de vídeo doméstico y profesional. También se dedicará una especial atención a los elementos de hardware y software necesarios para el trabajo en este interesante campo de aplicación del Amiga.

Además se incluirán temas de programación, pequeñas rutinas de utilidad, magias, etc. Y para completar ese número 3, se está elaborando una guía de hardware que prestará una gran ayuda a los usuarios que deseen ampliar su equipo con la gran oferta de nuestro mercado. ■

INFORMAT 89

La feria de Barcelona abrirá sus puertas desde el día cinco de junio hasta el día diez. Nuestra compañía, CW Communications a la que pertenecen Commodore World y Amiga World, estará presente en el palacio 2, nivel 1, stand 160.

A diferencia de otras ferias similares, Informat es un salón informático exclusivamente profesional. La ubicación del certamen es en los palacios 2 y 13 del recinto Fira de Barcelona. ■

MAD COMPUTER EN INFORMAT

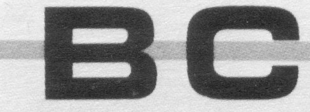
El stand que Commodore presenta cada año en Informat, suele estar apoyado con la presencia de varios distribuidores barceloneses. Este año, además, Mad Computer nos comunica su asistencia como especialista en Amiga. ■

LO USAN LAS TELEVISIONES DE TODO EL MUNDO

GENLOCK PAL



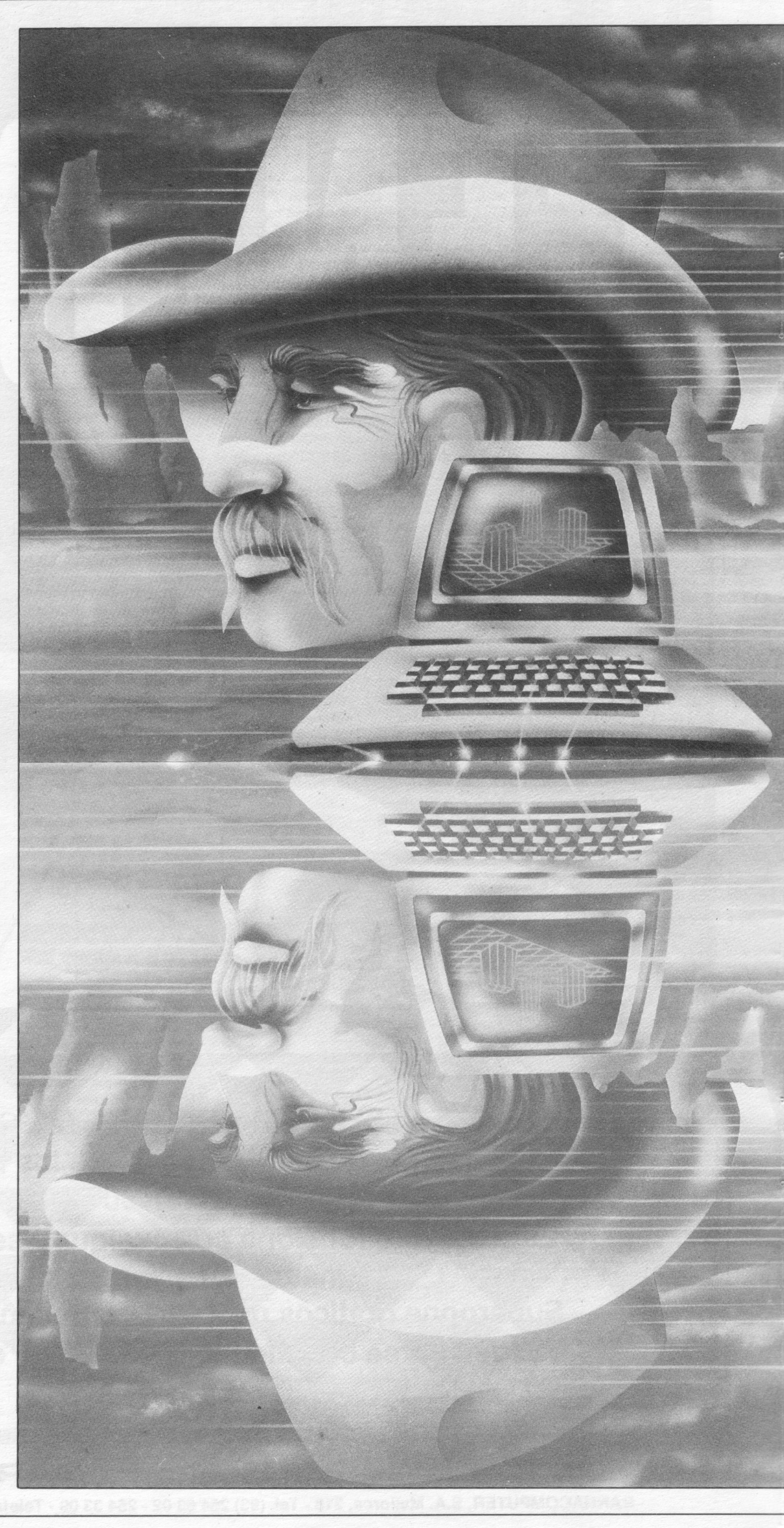
El video interface de *AMIGA*
que puede usarse con todos los modelos
(*AMIGA* 500-1000-2000)
Este aparato funciona con cualquier señal de video, magnetoscopio y discos laser
Superpone gráficos de *AMIGA* con señales de video.
Puede usarse con monitores de video de bajo precio



BARNACOMPUTER

BARNACOMPUTER, S.A. Mallorca, 218 - Tel. (93) 254 63 02 - 254 33 09 - Telefax (93) 254 5623 - 08008 Barcelona

Es probable que muchos de vosotros soñéis alguna vez con ser un gran programador de juegos, con tener mucho éxito, trabajar en una prestigiosa casa, etc. Pero también es casi seguro que pensándolo mejor, desistáis por no saber qué clase de «budú» sería necesario a la hora de crear un juego. Pues bien, en esta serie de artículos, os podréis enterar de los trucos, herramientas de trabajo, y «nuevas tecnologías» en la creación de juegos. Un ejemplo práctico es la conversión de un juego comercial de Spectrum a Commodore, cuyo resultado final será una demo jugable (parte del juego total).



ESPEJO

Por Francisco Javier Rodríguez

El primer problema que se me presentó fue cómo pasar los gráficos de Spectrum a Commodore. Por el RS232 sería muy fácil si ellos tuviesen un equipo Commodore, pero no es así. Tras darle muchas vueltas al asunto me di cuenta de que no era tan difícil, sólo era necesario tener un par de programas que cualquiera puede tener: el Simulador de Spectrum (comercial) y el SPOILER (editado en los discos Aplicaciones III). Después de hacerme con este par de programas, procedí a cargar el Simulador de Spectrum y tras tener éste en la memoria, cargué los gráficos de los personajes de una cinta de Spectrum (comprobé con satisfacción el buen resultado). Cuando los gráficos estaban en memoria procedí a hacer un reset; a partir de aquí fue donde intervino el Spoiler, programa que esta revista publicó en el disco Aplicaciones III. Con este programa se puede ver todo el contenido de la memoria en formato carácter o en formato sprite, además también puedes modificarlos, crearlos de nuevo, volcar cualquier zona de memoria a disco, etc. Lo primero que hice fue buscar por toda la memoria los personajes, en formato sprite, y una vez encontrados procedí a ajustarlos para que del formato en que estaban en la memoria de Spectrum pasaran a formato de Sprite.

Después de volcar la memoria de los personajes a disco, me dí cuenta de que serían necesarios tres sprites para cada personaje, y como en la pantalla deberían aparecer tres «malos» además de la protagonista, tendría que recurrir a la técnica de los multi-sprites, técnica que habréis visto más de una vez en esta revista y que no creo os represente problema alguno.

Una vez creada la rutina multi-sprite, había que crear otra para el movimiento de los personajes. Cada movimiento completo de un personaje, a derecha o izquierda, consta de ocho secuencias diferentes, por lo cual la memoria total que iban a ocupar los gráficos del juego era de 10k. No suponía ningún problema lo de la memoria, ya que si en un Spectrum había cabido todo, en un Commodore cabría mejor. Pero aun así, me puse a trabajar en una rutina que se llama «Espejo» (crear una imagen simétrica a la que tenemos en memoria). Con esta rutina la memoria total que iban a ocupar los gráficos quedaría en la mitad (5k), ya que sólo es necesario tener en memoria las ocho secuencias de todos los personajes hacia la derecha. Y en el caso de que cualquier personaje cambie la dirección hacia la izquierda, esta rutina girará las secuencias del personaje en cuestión. Para que lo entendáis, adjunto la rutina «Espejo».

Habréis visto cómo a la hora de crear los personajes de un juego, no es necesario hacerlo en todas las direcciones con sus secuencias, sino que basta con hacerlo en una sola dirección y crear la rutina que desde el juego «dé la vuelta» al personaje. La rutina para crear el «espejo» de poner el personaje «patas arriba» os la dejo a vosotros (es facilita).

Volviendo al tema, y si ya habéis copiado el programa, lo habéis ejecutado y jugado con él, os contaré que me di cuenta de que la rutina no serviría para el personaje principal. El problema es la velocidad, tarda unos dos segundos en girar veinticuatro secuencias de sesenta y cuatro bytes de longitud. Mientras crea las imágenes, utilizando multi-sprites, el sprite que duplicaba desaparecía de la pantalla por ese espacio de tiempo. Además de ser el personaje principal el que necesita mayor rapidez de respuesta en lo que se refiere al cambio de dirección.

En lo referente al resto de personajes serviría, ya que la dirección que van a tomar los malos se decide antes de que éstos aparezcan en la pantalla, y además los dos segundos que tarda la rutina, en el caso de que éstos tengan que ser girados, servirá para que no aparezcan inmediatamente después de la protagonista en un cambio de pantalla (maté dos pájaros de un tiro).

El movimiento del personaje principal, decidí hacerlo mediante «tablas», siendo éste el método más rápido. Por si hay alguien que no se-

Cada movimiento completo de un personaje, a derecha o izquierda, consta de ocho secuencias diferentes, por lo cual la memoria total que iban a ocupar los gráficos del juego era de 10k.

El movimiento del personaje principal decidí hacerlo mediante «tablas», siendo éste el método más rápido.

pa de qué va el tema, se trata de lo siguiente: las secuencias del personaje hacia la derecha, menos los diferentes tipos de golpes, se encuentran bajo ROM a partir de las direcciones A528, A648, A768, A888, A9A8, AAC8, ABE8, AD08. La tabla correspondiente sería la siguiente:

```
.: 1000 A5 28 A6 48 A7 68 A8 88
.: 1008 A9 A8 AA C8 AB E8 AD 08
```

La demo de la rutina «Espejo» usa este método para el movimiento del personaje cuando no tiene que ser girado, y podéis comprobar la velocidad con la que se muestran las diferentes secuencias.

Tras estas breves explicaciones sobre los listados, paso a comentaros más extensamente la rutina «ESPEJO» así como la rutina «Joy», que es la que realmente decide si tiene que crear las imágenes simétricas de los personajes.

Empezando por la rutina «JOY» lo único que es importante son las comparaciones del joystick a derecha o izquierda que es donde se decide si tiene que ir a la rutina «ESPEJO».

La rutina utiliza un byte cuyo uso es exclusivamente de «flag». En este byte estará activado el bit. n.º 1 si el último movimiento fue a la derecha , y el bit n.º 2 si el movimiento fue a la izquierda (byte «flag» = $1127).

Como podéis comprobar, en las direcciones comprendidas de $1056 a $1081 es donde aparece este byte. En la dirección $1056 es donde se compara si la dirección del joystick fue hacia la derecha. Si no fue hacia la derecha, saltará hacia la dirección $10E6 donde se comprobará si fue a la izquierda.

Suponiendo que la dirección fuese hacia la derecha, lo primero que hace es tomar el valor byte «flag» y compararlo con el valor uno (recordad que el uno, significa que el anterior movimiento fue a la derecha), si el resultado es diferente de cero quiere decir que no hace falta ir a la rutina «ESPEJO», pero en el caso de que el resultado fuese cero querría decir que el valor del byte «flag» era dos (recordad que el bit 2.º activado quiere decir que el anterior movimiento fué a la izquierda), por lo cual tendrá que crear las imágenes de nuevo hacia la derecha (rutina «espejo»), pero antes de que acceda a la rutina donde las creará, primero borra el bit 2º del byte «flag» y activa el bit 1.º del mismo byte como podéis ver en las direcciones comprendidas entre $1061 y $1068; saltando después a la rutina «espejo». Los pasos seguidos en la comprobación del joystick hacia la izquierda $10E6 son los mismos, comparándose el byte «flag» son dos.



Empezando por la rutina «JOY» lo único que es importante son las comparaciones del joystick a derecha o izquierda que es donde se decide si tiene que ir a la rutina «ESPEJO».

Una vez dentro de la rutina «ESPEJO» (situada a partir de $1084) se sigue el siguiente proceso. Primero activa la RAM que hay bajo ROM, ya que es donde están almacenados los datos de los personajes, seguidamente salta a la rutina de preparación dónde se asignan los valores de diferentes bytes cuya función será la siguiente:

$110E — byte de comparación. Este será el byte correspondiente de ir comparando bit a bit, los bits que están activados en el byte que hay que girar.

$110F — Este es el byte que se encargará de ir activando los correspondientes bits en el byte buffer (la función de estos dos bytes se verá después más detalladamente).

$1110 — Esta es la posición que guardará el byte ya girado (byte buffer).

$1111 — Este es el byte contador, el cual se encargará de repetir el proceso de comparación y activación ocho veces, ya que son ocho los bits que tiene un byte.

$1011 — Aquí estará el incremento para la tabla de las diferentes direcciones donde se encuentran los datos del personaje (a partir de $1000).

$1128 — Es un contador de tres bytes, o una fila de un sprite.

$1129 — Es el contador de los 63 bytes que tiene un Sprite o 21 filas de 3 bytes cada una.

A continuación se carga en un puntero ($FB+$FC) la dirección de la primera secuencia (acordaos, de que las direcciones de las ocho secuencias se encuentran a partir de la dirección $1000 en forma de byte alto y byte bajo). En caso de que el valor que se encuentre en la tabla de direcciones sea un cero (marca que uso como final de secuencias que hay que hacer), se saldrá de la rutina ya que no quedan más secuencias por girar (una secuencia son los tres sprites de los que está compuesto el personaje).

Una vez cargada la dirección de la secuencia en el puntero, se va a la rutina encargada de girar un byte, la cual se encuentra a partir de la dirección $112B que explico a continuación.

Primero cargo el acumulador con el valor contenido en la dirección base (dirección que contiene el puntero) más el valor de Y; por lo que si la dirección base (db) fuese A428 y el registro Y tuviese el valor tres, la direción final de la que se toma el valor será A42B.

Una vez que el registro contiene el dato de la dirección, se comprueba si el primer bit está activado (acordaos

del byte de comparación), si éste no estuviese activado se comprobaría el segundo, tras decrementar el contador así como mover el contenido del byte de comparación un bit a la izquierda, mover el contenido del byte de activación un bit a la derecha y repetir el proceso desde el principio. En el caso de que el bit de la posición con el que es comparado se encuentre activado. El siguiente paso es el de activarlo en el byte buffer ($1110) pasando a mover el contenido del byte de comprobación un bit a la izquierda, y el byte de activación un bit a la derecha, prosiguiendo con el siguiente bit hasta haber comprobado los ocho del byte.

Ejemplo:

byte comprobación	00000001
byte activación	10000000
byte buffer	00000000
byte contador	00001000
dato a comprobar	01010101

Primer paso.

```
ini     lda 01010101
        and 00000001
        beq nbit
sbit    lda 00000000
        ora 10000000
        sta 00000000 = 10000000
nbit    asl 00000001 = 00000010
        lsr 10000000 = 01000000
        dec 00001000 = 00000111
        bne ini
        rts
```

Segundo paso.

```
ini     lda 01010101
        and 00000010
        beq nbit
sbit    lda 10000000
        ora 01000000
        sta 10000000
nbit    asl 00000010 = 00000100
        lsr 01000000 = 00100000
        dec 00000111 = 00000110
        bne ini
        rts
```

En este ejemplo podéis ver cómo funciona la rutina. Si se continúan haciendo los siguientes pasos veréis cómo el byte resultado (ya girado) pasaría de ser 01010101 a 10101010, con lo cual ya estaría hecho el espejo (o imagen simétrica) de uno de los 63 bytes que tiene un sprite.

Una vez retornase de la rutina antes explicada, el programa seguiría con la lectura del contador de bytes en una fila (cbf), si éste fuera el n.º 2 (el cbf se incrementa, comparando los bytes de izquierda a derecha. Por lo cual, cuando éste es un dos quiere decir que es el primer byte de la fila, cuando es un 1 es el 2.º, y cuando es 0 es el 3.º), se guardaría el valor del byte buffer en la pila, a continuación el valor del registro Y (recordad que es el incremento sobre la db). A continuación, se incrementa el cbf y finalmente salta a la dirección $10E4 cuya función y la de las

```
LISTADOS PARA EL ARTICULO "SABRINA"

*** LISTADO 1. ***
          .:1000 A5 28 A6 48 A7 68 A8 88
          .:1008 A9 A8 AA C8 AB E8 AD 08

*** LISTADO 2. ***
      C000  SEI          ; inicio rutina multi-spr
      C001  LDA #$0D     ; preparación interrupción
      C003  STA $07F8    ; y sprites
      C006  LDA #$64
      C008  STA $07F9
      C00B  LDA #$64
      C00D  STA $D000
      C010  STA $D001
      C013  STA $D002
      C016  LDA #$79
      C018  STA $D003
      C01B  LDA #$00
      C01D  STA $D027
      C020  STA $D028
      C023  LDA #$03
      C025  STA $D015
      C028  LDA #$8D
      C02A  STA $D012
      C02D  LDA #$81
      C02F  STA $D01A
      C032  LDA $D011
      C035  AND #$7F
      C037  STA $D011
      C03A  LDA #$46
      C03C  STA $0314
      C03F  LDA #$C0
      C041  STA $0315
      C044  CLI
      C045  RTS
      C046  LDA $D019    ; interrupción por barrido?
      C049  STA $D019
      C04C  AND #$01
      C04E  BNE $C053    ; ¿si? vete a $C053
      C050  JMP $EA31    ; sino interrupciones normales
      C053  LDA $D012    ; linea del raster
      C056  CMP #$00     ; es la primera
      C058  BMI $C06C    ; bifurca sólo si es mayor de 128
      C05A  LDA #$8D     ; sino carga nueva linea de raster
      C05C  STA $D012
      C05F  LDA #$64     ; y vuelve a poner el sprite
      C061  STA $D001    ; duplicado en su sitio
      C064  LDA #$0D     ; con su imagen
      C066  STA $07F8
      C069  JMP $C07B    ; y vete a leer joystick
      C06C  LDA #$00     ; si es mayor de 128 carga
      C06E  STA $D012    ; linea inicial
      C071  LDA #$8E     ; y duplica sprite
      C073  STA $D001
      C076  LDA #$0F     ; con su nueva imagen
      C078  STA $07F8
      C07B  SEI          ; inhibe interrupciones
      C07C  TSX          ; transfiere estado de la pila
      C07D  STX $4000    ; a X y guardalo
      C080  JSR $1046    ; vete a rutina joy
      C083  LDX $4000    ; recupera estado de la pila
      C086  TXS          ; y transfierelo
      C087  CLI          ; habilita interrupciones
      C088  JMP $FEBC    ; y sal de interrupción por barrido

TABLA PUNT.  1000  A5 28 A6 48 A7 68 A8 88  ; tabla de imágenes
             1008  A9 A8 AA C8 AB E8 AD 08  ; para Sabrina
             1010  00 00

MOV          1017  LDA #$75     ; habilitar RAM bajo ROM
             1019  STA $01
             101B  LDX $112A    ; cargar contador secuencias
             101E  LDY #$00     ; inicializar Y
             1020  LDA $1000,X  ; cargar byte alto dirección
             1023  CMP #$00     ; es cero?
             1025  BNE $102B    ; no, pues bifurca nueva dirección
             1027  TAX          ; si, X=A
             1028  CLC          ; salto forzoso a la salida
             1029  BCC $103E    ; de la rutina
             102B  STA $FC      ; carga byte alto
             102D  INX          ; X=X+1
             102E  LDA $1000,X  ; carga byte bajo y...
             1031  STA $FB      ; guardalo en puntero
             1033  LDA ($FB),Y  ; valor de la dirección+Y
             1035  STA $0340,Y  ; y a bloque imagen de sprite
             1038  INY          ; Y=Y+1
             1039  CPY #$BD     ; Y=BD
             103B  BNE $1033    ; no, pues siguiente byte
             103D  INX          ; si, incrementa X
             103E  STX $112A    ; y guardalo
             1041  LDA #$77     ; habilitar de nuevo la ROM
```

```
          1043  STA $01
          1045  RTS              ; y salir de la rutina.
JOY       1046  LDA $DC00        ; joystick en port-2
          1049  CMP #$7F         ; ningún valor
          104B  BNE $104E        ; si, ¿qué dirección?
          104D  RTS              ; no, vuelve
          104E  CMP #$6F         ; ¿disparo?
          1050  BNE $1056        ; no, otra dirección
          1052  JSR $2001        ; si, vete a rutina disparo
          1055  RTS              ; y vuelve
          1056  CMP #$77         ; ¿derecha?
          1058  BNE $106E        ; no, otra dirección
          105A  LDA $1127        ; si, última dirección de Sabrina
          105D  AND #$01         ; ¿fué a la derecha?
          105F  BNE $1017        ; si, no hace falta hacer espejo
          1061  LDA $1127        ; no, entonces poner el flag
          1064  AND #$FC         ; de dirección hacia la derecha
          1066  ORA #$01         ;
          1068  STA $1127        ; y guardalo para próxima vez
          106B  CLC              ; salto forzoso rutina
          106C  BCC $1084        ; Espejo
          106E  CMP #$7B         ; ¿izquierda?
          1070  BEQ $1073        ; si, bifurca dirección
          1072  RTS              ; dirección no permitida, vuelve.
          1073  LDA $1127        ; último movimiento
          1076  AND #$02         ; ¿fué a la izquierda?
          1078  BNE $1017        ; si, no hace falta hacer espejo
          107A  LDA $1127        ; no, entonces poner el flag
          107D  AND #$FC         ; de dirección hacia la izquierda
          107F  ORA #$02         ;
          1081  STA 1127         ; y guardarlo para próxima vez
                                 ; haz el Espejo

ESPEJO    1084  LDA #$75         ; habilitar RAM bajo ROM
          1086  STA $01          ;
          1088  JSR $1112        ; rutina de preparación
          108B  LDA #$00         ; inicializa contador
          108D  STA $1011        ; de secuencias
```

siguientes direcciones comentaremos después.

Si el cbf fuera cero, se recuperaría el último valor que metimos en la pila (en este caso el incremento de la db), por lo que volveríamos a estar en el primer byte de la fila, y guardaríamos el valor del byte buffer en la dirección contenida en el puntero más el incremento que acabamos de recuperar; a continuación, se toma el primer valor que habíamos metido en la pila (valor del byte buffer cuando estaba en el primer byte de la fila) e incrementamos el registro Y dos veces para que el incremento sobre db vuelva a ser el mismo que tenía cuando estaba en el byte tercero de la misma fila, guardando el byte buffer del byte primero en la dirección del byte segundo. Una vez hecho esto, se inicializa el cbf a dos y se salta a la dirección $10E4. Cuando el valor cbf es uno, lo único que se hace es guardar el valor del byte buffer en la db+Y.

¿Por qué hay que hacer los intercambios del byte buffer entre el byte primero y el byte segundo de una fila? Esto es debido a que un sprite tiene tres bytes (o caracteres) por fila, y para crear la imagen simétrica del sprite hay que cambiar (como se puede ver

```
        1090  LDA #$02       ; inicializa varios bytes
        1092  STA $1128      ; como diferentes contadores
        1095  LDA #$3F
        1097  STA $1129
        109A  LDX $1011      ; X=valor de contador de secuencias
        109D  LDY #$00
        109F  LDA $1000,X    ; byte alto de dirección
        10A2  CMP #$00       ; ¿es cero?
        10A4  BEQ $1101      ; si, salir rutina
        10A6  STA $FC        ; no, cargalo en puntero
        10A8  INX
        10A9  LDA $1000,X    ; byte bajo de dirección
        10AC  STA $FB        ; carga en puntero
        10AE  JSR $112B      ; rutina de espejo en un byte
        10B1  LDA $1128      ; ¿qué nº de byte es en la fila?
        10B4  CMP #$02       ; ¿es el primero?
        10B6  BNE $10C4      ; no, bifurca nueva dirección
        10B8  LDA $1110      ; si, carga byte resultado de buffer
        10BB  PHA            ; y guardalo en la pila
        10BC  TYA            ; guarda también el incremento
        10BD  PHA            ; sobre dirección base en la pila
        10BE  DEC $1128      ; decrementa contador de bytes
        10C1  CLC            ; salto forzoso a nueva dirección
        10C2  BCC $10E4
        10C4  CMP #$00       ; ¿es el tercero?
        10C6  BNE $10DC      ; no, bifurca nueva dirección
        10C8  PLA            ; si, recupera Y de byte primero
        10C9  TAY            ; en el incremento de dir. base
        10CA  LDA $1110      ; carga byte resultado de espejo
        10CD  STA ($FB),Y    ; y pokéalo
        10CF  PLA            ; recupera byte resultado byte 1º
        10D0  INY            ; e incrementa hasta byte 3º de la
        10D1  INY            ; misma fila de un bloque
        10D2  STA ($FB),Y    ; y pokéalo
        10D4  LDA #$02       ; inicializa contador de bytes
        10D6  STA $1128      ; de una misma fila
        10D9  CLC            ; salto forzoso a nueva dirección
        10DA  BCC $10E4
        10DC  LDA $1110      ; forzosamente es byte 2º
        10DF  STA ($FB),Y    ; este se queda en el mismo sitio
        10E1  DEC $1128      ; pero también girado,dec cont.bytes
        10E4  JSR $1112      ; preparación de bytes
        10E7  DEC $1129      ; decrementa contador de bloques
        10EA  BNE $10F2      ; no se han hecho 63 bytes, continúa
        10EC  INY            ; si, incrementa Y
        10ED  LDA #$3F       ; e inicializa contador de bloques
        10EF  STA $1129      ; a 63 bytes
        10F2  INY            ; incrementa Y
        10F3  CPY #$BD       ; ¿es 189?,o sea tres bloques
        10F5  BCC $10AE      ; si es menor o igual sigue...
        10F7  INX            ; sino incrementa contador de
        10F8  STX $1011      ; secuencias
        10FB  LDX #$00       ; y pon la pila
        10FD  TXS            ; a partir de $0100
        10FE  CLC            ; sigue con siguiente secuencias
        10FF  BCC $1090      ;
        1101  TAX            ; apuntador de pila a $0100
        1102  TXS            ; acabadas todas las secuencias
        1103  LDA #$C0       ; creo mi propia dirección de
        1105  PHA            ; retorno
        1106  LDA #$82
        1108  PHA
        1109  LDA #$77       ; activo la ROM de nuevo, y
        110B  STA $01
        110D  RTS            ; vuelvo a Multi-spr

bytes espejo:  110E  01 80 00 08

Rut pre. 1112 LDA #$01       ;byte de comparación de bits
         1114 STA $110E
         1117 LDA #$80       ;byte de activación de bits
         1119 STA $110F
         111C LDA #$00       ;byte buffer resultado final
         111E STA $1110
         1121 LDA #$08       ;contador de 8 bits (1 byte)
         1123 STA $1111
         1126 RTS            ;retorna

bytes contadores:  1127 01 02 3F 00

espej.1  112B LDA ($FB),Y    ; carga byte de dirección base+Y
         112D AND $110E      ; bit activado
         1130 BEQ $113B      ; no, bifurca nueva dirección
         1132 LDA $1110      ; si, carga byte buffer
         1135 ORA $110F      ; y activa correspondiente bit
         1138 STA $1110      ; y dejarlo en byte buffer
         113B ASL $110E      ; siguiente bit comparación
         113E LSR $110F      ; siguiente bit activación
         1141 DEC $1111      ; decrementa contador de bits
         1144 BNE $112B      ; no se ha completado byte
         1146 RTS            ; ¿si?, pues vuelve.
```

en el dibujo) lo que antes estaba en el byte primero (siempre de izquierda a derecha) al tercero y viceversa. Así tenemos hecho el espejo de una de las 21 filas que tiene un sprite, y sólo tendríamos que repetir este proceso otras veinte veces para un sprite.

Seguimos el comentario de las rutinas en la dirección $10E4, que como podéis comprobar es donde se manda-ba tras acabar las operaciones de cbf. Aquí lo primero que se hace es ir de nuevo a la rutina de preparación (recordad que esta rutina está a partir de $1112). El siguiente paso (dirección $10E7) es decrementar la posición encargada de contar los 63 bytes que tiene un sprite ($1129). Si este contador ha llegado a cero, se incrementa el registro Y para que la siguiente dirección de la que se toma un dato, sea la primera de la 1.ª fila del siguiente sprite, a continuación se vuelve a inicializar este contador a 63.

Si el contador de 63 bytes no llegase todavía a cero, se incrementa el registro Y y se compara con el valor 193 (equivalente a los bytes suma de tres sprites+1). Si no fuese así, se seguiría con el mismo proceso hasta ahora explicado a partir de la dirección $ 1090 (podéis fijaros en el listado).

Si el registro Y tomase el valor de 193, querría decir que ya se ha acabado de hacer una secuencia de tres sprites, por lo que habría que seguir con la siguiente. Para ello lo primero que se hace es incrementar el registro X y bajar el estado de la pila a cero, con lo cual en las siguientes operaciones serán machacados los datos que ésta contenía (así, no se podrá hacer ningún programa basic). Este método es muy útil para la protección de los listados en c.m. ya que así no podrán ser pisados por ningún programa basic introducido por error.

Por último a partir de la dirección $1101 se encuentra el retorno de la rutina «ESPEJO» hacia la rutina de la cual fue llamada (RUTINA MULTI-SPR), se accede a esta dirección desde $109F, donde se comprobará si el dato tomado de la tabla era el cero, utilizando como final de secuencias. Una vez dentro de la dirección $1101 lo primero que hago es poner el estado de la pila de nuevo a cero y guardo en ella la dirección de retorno, ya que la que estaba guardada en la pila ha sido machacada. Después de esto vuelvo a activar la ROM y vuelvo a MULTI-SPR con el correspondiente RTS.

Espero que esta rutina os sea de utilidad en vuestros programas, siendo la rutina «ESPEJO» fácilmente modificable para el número de secuencias que queráis introducir, así como la cantidad de sprites que compondrán una secuencia 1, 2 ó 3 como máximo). ■

```
ANTES DE TECLEAR O CARGAR ESTE LISTADO
HAZ LOS SIGUIENTES POKES:

POKE44,32:POKE32*256,0:NEW

PROGRAMA: ESPEJO.DEMO          LISTADO 1

1 REM RUTINA ESPEJO                   .57
2 REM (C)1989 BY FRANCISCO J.RODRIG .164
UEZ
5 REM (C)1989 BY COMMODORE WORLD      .21
6 :                                  .238
10 REM GENERADOR DE LOS PROGRAMAS:   .154
11 REM MULTISPR V1.0 ($1000-$114A)    .53
12 REM SABRINA.MOV    ($A528-$AE28)    .8
13 REM ESPEJO V.10    ($C000-$C08B)   .65
15 :                                 .247
50 FORQ=49152TO49291:READA:POKEQ,A:  .64
S=S+A:NEXT
53 FORQ=42280TO44584:READA:POKEQ,A:  .83
S=S+A:NEXT
55 FORQ=4096TO4426:READA:POKEQ,A:S=  .15
S+A:NEXT
60 IFS<>217541THENSTOP                .34
70 PRINT"[CLR][5CRSRD]"              .162
75 FORI=1TO7:PRINT"[6SPC][RVSON][WH  .57
T][10SPC]":NEXT
80 PRINT"[2CRSRD]TECLEA SYS 49152 Y .120
 MUEVE EL JOYSTICK":END
99 :                                  .75
100 DATA 120,169,13,141,248,7,169     .48
101 DATA 14,141,249,7,169,100,141    .157
102 DATA 0,208,141,1,208,141,2       .134
103 DATA 208,169,121,141,3,208,169   .199
104 DATA 0,141,39,208,141,40,208     .124
105 DATA 169,3,141,21,208,169,141    .245
106 DATA 141,18,208,169,129,141,26   .200
107 DATA 208,173,17,208,41,127,141    .19
108 DATA 17,208,169,70,141,20,3      .238
109 DATA 169,192,141,21,3,88,96      .191
110 DATA 173,25,208,141,25,208,41    .210
111 DATA 1,208,3,76,49,234,173       .21
112 DATA 18,208,201,0,48,18,169      .214
113 DATA 141,141,18,208,169,100,141  .27
114 DATA 1,208,169,13,141,248,7      .250
115 DATA 76,123,192,169,0,141,18     .9
116 DATA 208,169,142,141,1,208,169   .4
117 DATA 15,141,248,7,120,186,142    .9
118 DATA 0,64,32,70,16,174,0         .122
119 DATA 64,154,88,76,188,254,0      .99
120 DATA 0,15,0,0,63,192,0           .60
121 DATA 126,224,0,123,64,0,247      .45
122 DATA 0,0,255,128,0,247,0         .4
123 DATA 1,255,0,1,236,0,3           .143
124 DATA 220,0,3,222,0,1,191         .78
125 DATA 0,3,187,192,3,189,96        .233
126 DATA 7,122,160,4,117,64,0        .128
127 DATA 123,128,0,100,0,0,234       .209
128 DATA 0,0,196,0,0,213,0           .12
129 DATA 0,0,203,128,0,185,192       .135
130 DATA 0,184,96,1,148,48,1         .4
131 DATA 170,56,1,149,8,1,171        .233
132 DATA 0,1,94,0,0,62,0             .16
133 DATA 0,95,0,0,95,0,0             .247
134 DATA 111,0,0,111,0,0,119         .162
135 DATA 128,0,115,128,0,113,128     .129
136 DATA 0,225,192,1,225,192,1       .172
137 DATA 193,192,3,128,192,3,0       .135
138 DATA 192,6,6,0,192,20,0          .244
139 DATA 96,24,0,64,38,0,176         .147
140 DATA 6,0,88,3,0,76,0             .250
141 DATA 0,0,0,0,0,0,0               .179
142 DATA 0,0,0,0,0,0,0               .180
143 DATA 0,0,0,0,0,0,0               .181
144 DATA 0,0,0,0,0,0,0               .182
145 DATA 0,0,0,0,0,0,0               .183
146 DATA 0,0,0,0,0,0,0               .184
```

```
147 DATA 0.0,255,0,255,0,255          .155
148 DATA 0,255,0,255,0,255,0          .214
149 DATA 255,0,255,0,255,0,255        .17
150 DATA 0,255,0,255,0,231,127        .248
151 DATA 0,255,0,255,0,255,0          .217
152 DATA 255,0,255,0,255,0,255        .20
153 DATA 0,255,0,255,0,255,0          .219
154 DATA 255,0,255,0,255,0,255        .22
155 DATA 0,255,0,255,0,255,0          .221
156 DATA 255,0,255,0,255.0,255        .24
157 DATA 0,255,0,255,0,255,0          .223
158 DATA 255,0,255,0,255,0.255        .26
159 DATA 0,255,0,255,0,255,0          .225
160 DATA 255,0,255,0,255,0,255        .28
161 DATA 0,0,30,0,0,63,224            .245
162 DATA 0,125,192,0,119,0,0          .78
163 DATA 255,0,1,247,128.3.239        .91
164 DATA 0,1,255,0,1,236,0            .244
165 DATA 3,220,0,1,222,0.3            .67
166 DATA 191,128,7,191,224,3,189      .230
167 DATA 80,3,58,176,2,57,96          .81
168 DATA 0,55,128,0,114,0,0           .252
169 DATA 116.0,0,108,0,0,100          .5
170 DATA 0,0,0,106,0,0,91             .116
171 DATA 0.0,93,0,0,101,192           .131
172 DATA 0,106,192,0,101,64,0         .134
173 DATA 106,128,0,149,0.0,62         .235
174 DATA 0,0,31,0,0,31,0              .210
175 DATA 0.15,0,0,15,0,0              .159
176 DATA 7,0,0,7,0,0,27               .86
177 DATA 0,0,59,0,0,51,128            .235
178 DATA 0,99,128,0,67,128.2          .60
179 DATA 129.192.3.3,129,128,4        .31
180 DATA 193.0,0,193,128,0,98         .154
181 DATA 192.0.1,96,0,1.48            .123
182 DATA 0.0,0,0,0,0,0                .220
183 DATA 0.0,0,0,0,0.0                .221
184 DATA 0.0.0.0.0,0,0                .222
185 DATA 0.0,0.0.0.0,0                .223
186 DATA 0,0,0,0,0,0,0                .224
187 DATA 0,0.0,0,0,0,0                .225
188 DATA 0,0,0,0,255,0,255            .50
189 DATA 0.255.0,255,0,255,0          .255
190 DATA 255,0,255,0,255,0,255        .58
191 DATA 0,255,0,255,0,255,0          .1
192 DATA 255,0,255,0,255,0,255        .60
193 DATA 0,255,0,255,0,255,0          .3
194 DATA 255,0,255.0,255,0,255        .62
195 DATA 0,255,0,255,0,255,0          .5
196 DATA 255,0,255,0,255,0,255        .64
197 DATA 0,255,0,255,0,255,0          .7
198 DATA 255.0,255,0,255,0.255        .66
199 DATA 0,255,0,255,0,255,0          .9
200 DATA 255,0,255,0,255,0,255        .68
201 DATA 0,255,0,255,0,255,0          .11
202 DATA 255.0,0,62,0,0,255           .144
203 DATA 192,1,255,192,1,250,128      .175
204 DATA 2,247,0,1,255.128,3          .20
205 DATA 247,0,7.255,0,3,236          .131
206 DATA 0.7,222,0,7,223,128          .192
207 DATA 11,159,224,7,158,176,23      .149
208 DATA 61,80,14,58,160,8,57         .78
209 DATA 192,0,59,0.0,61,0            .185
210 DATA 0,50,0,0,54,0,0              .32
211 DATA 50,0,0,0,52,0,0              .43
212 DATA 54,0,0,22,0,0,89             .198
213 DATA 0,0,154,0,1.89,0             .3
214 DATA 2.170,0,1,85,0,0             .128
215 DATA 62,0,0,62.0,0,30             .147
216 DATA 128,0,30,192,0,30,224        .214
217 DATA 0,30,224,0,14,192,0          .55
218 DATA 13,0,0,28,0,0,88             .200
219 DATA 0.0,184,0,1,56,0             .133
220 DATA 0,48,0,0,0,48,0              .194
221 DATA 0,32,0,0,80,0,0              .27
222 DATA 240,0,0,88,0,0,76            .44
223 DATA 0,0,0,0,0,0,0                .5
224 DATA 0,0,0,0,0,0,0                .6
225 DATA 0,0,0,0,0,0,0                .7
226 DATA 0.0,0,0,0,0,0                .8
227 DATA 0,0,0,0,0,0,0                .9
228 DATA 0,0,0,0,0,0,0                .10
229 DATA 0,0.0,0,0,255,0              .155
230 DATA 255,0,255,0,255,0,255        .98
231 DATA 0,255,0,255,0,255,0          .41
232 DATA 255,0,255,0,255,0,255        .100
233 DATA 0,255,0,255,0,255,0          .43
234 DATA 255,0,255,0,255,0,255        .102
```

```
235 DATA 0,255,0,255,0,255,0          .45
236 DATA 255,0,255,0,255,0,255        .104
237 DATA 0,255,0,255,0,255,0          .47
238 DATA 255,0,255,0,255,0,255        .106
239 DATA 0,255,0,255,0,255,0          .49
240 DATA 255,0,255,0,255,0,255        .108
241 DATA 0,255,0,255,0,255,0          .51
242 DATA 255,16,0,255,0,255,0         .238
243 DATA 255,0,255,0,30.64,0          .247
244 DATA 63,192,0,254,192,1,251       .176
245 DATA 0,2,255,0,1,247,128          .155
246 DATA 3,255,0,3,247,0,1            .220
247 DATA 236,0,1,238,0,3,223          .67
248 DATA 0,3,223,192,5,158,224        .250
249 DATA 3,29,80,3,28,160,6           .63
250 DATA 29,192.0,13,0,0,45           .24
251 DATA 0,0,45,0,0,14,0              .165
252 DATA 0,22,0,0,0,22,0              .156
253 DATA 0,26,0,0,42,0,0              .29
254 DATA 83,0,2,171,128,1,85          .146
255 DATA 128,0,170,128,0,125,0        .23
256 DATA 0,62,128,0,30,0,0            .191
257 DATA 30,0,0,62,128,0,61           .218
258 DATA 192,0,61,192,0,59,128        .235
259 DATA 0,59,0.0,119,0,0             .240
260 DATA 118,0,0,230,0,0,236          .209
261 DATA 0,0,204,0,0,1,152            .174
262 DATA 0,1,60,0,6,150,0             .105
263 DATA 15,3,0,5,128,0,4             .240
264 DATA 192,0,0,0,0,0,0              .213
265 DATA 0,0,0.0,0,0,0                .48
266 DATA 0,0,0,0,0,0,0                .49
267 DATA 0,0,0,0,0,0,0                .50
268 DATA 0,0,0,0,0,0,0                .51
269 DATA 0,0,0,0,0,0,0                .52
270 DATA 0,0,0,0.0,0,255              .215
271 DATA 0,255,0,255,0,255,0          .82
272 DATA 255,0,255,0,255,0,255        .141
273 DATA 0,255,0,255,0,255,0          .84
274 DATA 255,0,255,0,255,0,255        .143
275 DATA 0,255,0,255,0,255,0          .86
276 DATA 255,0,255,0,255,0,255        .145
277 DATA 0,255,0,255,0,255,0          .88
278 DATA 255,0,255,0,255,0,129        .179
279 DATA 255,0,255,0,255,0,255        .148
280 DATA 0,255,0,255,0,255,0          .91
281 DATA 255,0,255,0,255,0.255        .150
282 DATA 0,255,0,255,0,255,0          .93
283 DATA 255,0,255,0,255.0,255        .152
284 DATA 0,255.0.255,0,7,0            .85
285 DATA 0,63.224,0,126,224,0         .142
286 DATA 253.64,1,119,0,0,255         .179
287 DATA 128.1,247,0,2,255,0          .102
288 DATA 3,236,0,5,238,0,3            .101
289 DATA 222,0,3,223,128,7,222        .12
290 DATA 224,5,191,80,11,110,160      .29
291 DATA 2,127,192,4,110,0,0          .80
292 DATA 118,0,0,239,0,0,215          .211
293 DATA 0,0,203.0,0,0,189            .144
294 DATA 128,1,222,192,2,170,96       .235
295 DATA 1,85.112,2,170.176,0         .58
296 DATA 85,16,0,58,128,0,31          .157
297 DATA 0,0,30,0,0,62,0              .144
298 DATA 0,62.0,0,61,0,0              .201
299 DATA 61,0,0,59,0.0,123            .170
300 DATA 0,0,119,0,0,227,0            .191
301 DATA 1,227,128,1,195.128,3        .74
302 DATA 131,128,3,1,0,0,6            .171
303 DATA 1,128,4,0,192,26,0           .60
304 DATA 128,28,1,224,38,0,176        .7
305 DATA 3,0,152,0,0,0,0              .138
306 DATA 0,0,0,0.0,0,0                .89
307 DATA 0,0,0,0,0,0,0                .90
308 DATA 0,0,0,0,0,0,0                .91
309 DATA 0,0,0.0,0,0,0                .92
310 DATA 0,0,0,0,0,0,0                .93
311 DATA 0,0,0,0,0,0,0                .94
312 DATA 255,0,255.0,255,0,255        .181
313 DATA 0,255,0,255,0.255,0          .124
314 DATA 255.0.255,0,255,0,255        .183
315 DATA 0.255,0,128,255,0,255        .150
316 DATA 0,255,0.255,0,255,0          .127
317 DATA 255.0,255,0,255,0.255        .186
318 DATA 0,255,0,255,0,255,0          .129
319 DATA 255,0,255,0,255,0,255        .188
320 DATA 0,255,0,255,0,255,0          .131
321 DATA 255,0,255,0,255,0,255        .190
322 DATA 0,255,0,255,0,255,0          .133
```

```
323 DATA 255,0,255,0,255,0,255          .192
324 DATA 0,255,0,255,0,255,0            .135
325 DATA 255,0,255,0,255,0,14           .200
326 DATA 0,0,31,64,0,127,224            .65
327 DATA 0,253,192,0,119,0,0            .214
328 DATA 255,128,1,119,0,1,255          .233
329 DATA 0,3,236,0,1,238,0              .222
330 DATA 1,222,0,3,223,0,3              .107
331 DATA 221,128,5,157,192,1,30         .106
332 DATA 224,3,29,80,0,46,160           .117
333 DATA 0,61,192,0,46,0,0              .8
334 DATA 118,0,0,78,0,0,0               .145
335 DATA 55,0,0,123,0,0,171             .216
336 DATA 0,0,85,0,2,169,128             .255
337 DATA 1,85,192,0,170,192,0           .36
338 DATA 63,64,0,31,0,0,31              .229
339 DATA 0,0,15,0,0,15,0                .252
340 DATA 0,7,0,0,7,0,0                  .147
341 DATA 14,0,0,30,0,0,29               .16
342 DATA 0,0,59,0,0,54,0                .135
343 DATA 0,102,0,3,70,0,0               .222
344 DATA 3,130,0,5,130,0,0              .61
345 DATA 197,0,0,239,0,0,5              .196
346 DATA 128,0,4,192,0,0,0              .205
347 DATA 0,0,0,0,0,0,0                  .130
348 DATA 0,0,0,0,0,0,0                  .131
349 DATA 0,0,0,0,0,0,0                  .132
350 DATA 0,0,0,0,0,0,0                  .133
351 DATA 0,0,0,0,0,0,0                  .134
352 DATA 0,0,0,0,0,0,0                  .135
353 DATA 255,0,255,0,255,0,255          .222
354 DATA 0,255,0,255,0,255,0            .165
355 DATA 255,0,255,0,255,0,255          .224
356 DATA 0,255,0,255,0,255,0            .167
357 DATA 255,0,255,0,255,0,255          .226
358 DATA 0,255,0,255,0,255,0            .169
359 DATA 255,0,255,0,255,0,255          .228
360 DATA 0,255,0,255,0,255,0            .171
361 DATA 255,0,255,0,255,0,255          .230
362 DATA 0,255,0,255,0,255,0            .173
363 DATA 255,0,255,0,255,0,255          .232
364 DATA 0,255,0,255,0,255,0            .175
365 DATA 255,0,255,0,255,0,255          .234
366 DATA 0,255,0,255,0,255,0            .177
367 DATA 15,0,0,63,192,0,125            .222
368 DATA 224,0,255,96,0,119,0           .119
369 DATA 0,255,128,1,247,0,1            .212
370 DATA 255,0,1,236,0,0,238            .41
371 DATA 0,0,222,0,1,223,0              .146
372 DATA 3,221,128,1,157,224,3          .35
373 DATA 62,176,1,29,80,0,58            .246
374 DATA 160,0,29,192,0,26,0            .27
375 DATA 0,28,0,0,26,0,0                .120
376 DATA 0,28,0,0,86,0,0                .217
377 DATA 215,0,0,90,0,0,153             .212
378 DATA 0,1,90,0,0,169,0               .137
379 DATA 1,86,128,0,191,0,1             .140
380 DATA 95,128,0,111,128,0,119         .99
381 DATA 192,0,59,192,0,61,224          .102
382 DATA 0,59,192,0,119,128,0           .21
383 DATA 111,0,0,92,0,0,48              .98
384 DATA 0,1,96,0,1,144,0               .157
385 DATA 0,2,208,0,0,208,0              .210
386 DATA 0,96,0,0,24,0,0                .165
387 DATA 44,0,0,38,0,0,0                .174
388 DATA 0,0,0,0,0,0,0                  .171
389 DATA 0,0,0,0,0,0,0                  .172
390 DATA 0,0,0,0,0,0,0                  .173
391 DATA 0,0,0,0,0,0,0                  .174
392 DATA 0,0,0,0,0,0,0                  .175
393 DATA 0,0,0,0,0,0,0                  .176
394 DATA 0,255,0,255,0,255,0            .205
395 DATA 255,0,255,0,255,0,255          .8
396 DATA 0,255,0,255,0,255,0            .207
397 DATA 255,0,255,0,255,0,255          .10
398 DATA 0,255,0,255,0,255,0            .209
399 DATA 255,0,255,0,255,0,255          .12
400 DATA 0,255,0,255,0,255,0            .211
401 DATA 255,0,255,0,255,0,255          .14
402 DATA 0,255,0,255,0,255,0            .213
403 DATA 255,0,255,0,255,0,255          .16
404 DATA 0,255,0,255,0,255,0            .215
405 DATA 255,0,j55,0,255,0,255          .18
406 DATA 0,255,0,255,0,231,126          .121
60138 DATA 0,255,0,255,0,255,0          .22
408 DATA 0,15,0,0,63,224,0              .219
409 DATA 126,224,0,251,64,0,247         .46
410 DATA 0,0,255,128,1,119,0            .167
411 DATA 1,255,0,1,236,0,0              .152
412 DATA 238,0,1,222,0,2,223            .87
413 DATA 0,1,221,192,1,221,96           .226
414 DATA 3,186,176,3,61,80,2            .185
415 DATA 59,224,0,53,0,0,50             .142
416 DATA 0,0,100,0,0,106,0              .153
417 DATA 0,0,100,0,0,40,0               .150
418 DATA 0,42,0,0,103,0,0               .11
419 DATA 107,0,0,213,128,0,234          .248
420 DATA 128,1,85,0,0,190,0             .85
421 DATA 1,127,0,0,63,0,0               .190
422 DATA 31,128,0,47,128,0,51           .87
423 DATA 192,0,57,192,0,57,192          .86
424 DATA 0,123,128,0,115,128,0          .241
425 DATA 227,0,0,231,0,0,198            .38
426 DATA 0,0,1,134,0,1,12               .101
427 DATA 0,2,156,0,7,150,0              .208
428 DATA 2,195,0,2,96,0,0               .193
429 DATA 0,0,0,0,0,0,0                  .212
430 DATA 0,0,0,0,0,0,0                  .213
431 DATA 0,0,0,0,0,0,0                  .214
432 DATA 0,0,0,0,0,0,0                  .215
433 DATA 0,0,0,0,0,0,0                  .216
434 DATA 0,0,0,0,0,0,0                  .217
435 DATA 0,0,255,0,255,0,255            .188
436 DATA 0,255,0,255,0,255,0            .247
437 DATA 255,0,255,0,255,0,255          .50
438 DATA 0,255,0,255,0,255,0            .249
439 DATA 255,0,255,0,255,0,255          .52
440 DATA 0,255,0,255,0,255,0            .251
441 DATA 255,0,255,0,255,0,255          .54
442 DATA 0,255,0,255,0,255,0            .253
443 DATA 255,0,231,126,0,255,0          .220
444 DATA 255,0,255,0,255,0,255          .57
445 DATA 0,255,0,255,0,255,0            .0
446 DATA 255,0,255,0,255,0,255          .59
447 DATA 0,255,0,255,0,255,0            .2
448 DATA 255,0,255,0,255,0,255          .61
449 DATA 0,255,165,40,166,72,167        .150
450 DATA 104,168,136,169,168,170,20 .221
0
451 DATA 171,232,173,8,0,0,169          .192
452 DATA 0,141,17,16,169,117,133        .71
453 DATA 1,174,44,17,160,0,189          .236
454 DATA 0,16,201,0,208,4,170           .145
455 DATA 24,144,19,133,252,232,189      .28
456 DATA 0,16,133,251,177,251,153       .73
457 DATA 64,3,200,192,189,208,246       .112
458 DATA 232,142,44,17,169,119,133      .25
459 DATA 1,96,173,0,220,201,127         .250
460 DATA 208,1,96,201,111,208,4         .189
461 DATA 32,1,32,96,201,119,208         .240
462 DATA 20,173,41,17,41,1,208          .147
463 DATA 182,173,41,17,41,252,9         .168
464 DATA 1,141,41,17,24,144,22          .51
465 DATA 201,123,240,1,96,173,41        .82
466 DATA 17,41,2,208,157,173,41         .157
467 DATA 17,41,240,9,2,141,41           .122
468 DATA 17,169,117,133,1,32,20         .17
469 DATA 17,169,0,141,17,16,169         .230
470 DATA 2,141,42,17,169,63,141         .189
471 DATA 43,17,174,17,16,160,0          .104
472 DATA 189,0,16,201,0,240,91          .247
473 DATA 133,252,232,189,0,16,133       .248
474 DATA 251,32,45,17,173,42,17         .159
475 DATA 201,2,208,12,173,18,17         .46
476 DATA 72,152,72,206,42,17,24         .111
477 DATA 144,32,201,0,208,20,104        .176
478 DATA 168,173,18,17,145,251,104      .55
479 DATA 200,200,145,251,169,2,141      .246
480 DATA 42,17,24,144,8,173,18          .211
481 DATA 17,145,251,206,42,17,32        .108
482 DATA 20,17,206,43,17,208,6          .55
483 DATA 200,169,63,141,43,17,200       .110
484 DATA 192,189,144,183,232,142,17 .145
485 DATA 16,162,0,154,24,144,143        .54
486 DATA 170,154,169,192,72,169,130 .179
487 DATA 72,169,119,133,1,76,18         .106
488 DATA 16,1,128,0,8,169,1             .75
489 DATA 141,16,17,169,128,141,17       .190
490 DATA 17,169,0,141,18,17,169         .13
491 DATA 8,141,19,17,96,1,2             .152
492 DATA 63,8,177,251,45,16,17          .63
493 DATA 240,9,173,18,17,13,17          .12
494 DATA 17,141,18,17,14,16,17          .105
495 DATA 78,17,17,206,19,17,208         .248
496 DATA 229,96,0,0,0,255,0             .183
```

NOVEDADES abc analog, s. a.
PARA ORDENADORES AMIGA

GVP

GREAT VALLEY PRODUCTS

A2000 - HARD CARD

Discos duros en tarjeta para AMIGA 2000

Con fantásticas características, **ejemplo: A2000-HARD CARD 80Q** con zócalo para EPROM de autoarranque v 1.3; DMA; 80 MB de capacidad; **64 Kb de memoria CACHE; 19/11 mseg. de tiempo de acceso** y 2MB/seg. de velocidad de transferencias de datos; controlador SCSI para otros periféricos, etc.

TARJETAS CONTROLADORAS SCCI/RAM MULTIFUNCION

Con controlador SCSI de disco duro y posibilidad de instalar 2MB DRAM, zócalos para EPROMS de autoarranque del A2000, velocidad de hasta 4MB/seg. conector externo SCSI para hasta 7 periféricos en cascada, etc.

DISCO DURO REMOVIBLE PARA A2000

44 MB de capacidad, 25 mseg. de acceso, tecnología Winchester, formato 5,25", 1/2 altura, 1,25 MB/seg., buffer de 8KB, etc. Incluye la EPROM de autoarranque.



COMPUTERSYSTEME GMBH

SYSTEM 2000

Discos duros externos, 20, 30, 40 y 60 MB, 30 mseg., para: • AMIGA 500 Y 1000 • AMSTRAD PPC 512 Y 640; PC 1512 Y 1640; PCW 8256, 8512 Y 9512 • COMPATIBLES PC • SCHNEIDER EURO PC PS/2-30.

SUPERCARDS

Discos duros en tarjeta, 20 y 30- MB con 60 mseg. y 40 y 60 MB con 30 mseg.; para compatibles PC y XT; con programas MULTIPART, BACK DISK y PARK.

HD PLUS/ATARI ST

20, 30 y 40 MB con 60 mseg.; 60, 80, 100 y 120 MB con 28 mseg.; hasta 16 particiones; AUTO PARKER, memoria CACHE rápida; 1,1 MB/seg., conectables en cascada, etc.

Distribuido por **abc analog, s. a.**

Santa Cruz de Marcenado, 31
28015 MADRID Tels. (91) 248 82 13

Télex: 44561 BAB CE
Fax: (341) 542 50 59

RUEGO ENVIEN INFORMACION SIN COMPROMISO A:
NOMBRE
DIRECCION
TELEFONO
ORDENADOR

MAYO 89

SPRITE SCANNER

3.ª Parte

El modo de operar con Super Sprite es el siguiente. En una cinta (o en un disco) debe estar preparada una versión binaria del programa. Para lograrlo debe usarse un monitor de memoria, como por ejemplo el M.E.S. o el H.E.S. Mon, o cualquier otro: se ejecuta el cargador BASIC del programa y con el monitor se salva la zona de memoria correspondiente.

Una vez realizada esta operación, puede cargarse el programa que se desee explorar y se ejecuta normalmente. Para explorar el programa primero debe salirse de él sin alterar (por lo menos no demasiado) la memoria actual de la máquina. La solución es el Reset, efectuado de cualquier manera posible, o el Freezer saliendo después al sistema.

Ahora ya podemos cargar el programa con LOAD «...»,1,1 (versión en cinta) o LOAD «...»,8,1 (versión en disco). Una vez cargado, se activa con SYS 49152. Podemos entonces ir desplazándonos por la memoria y viendo los sprites que el programa esconde, así como marcarlos para su posterior recuperación.

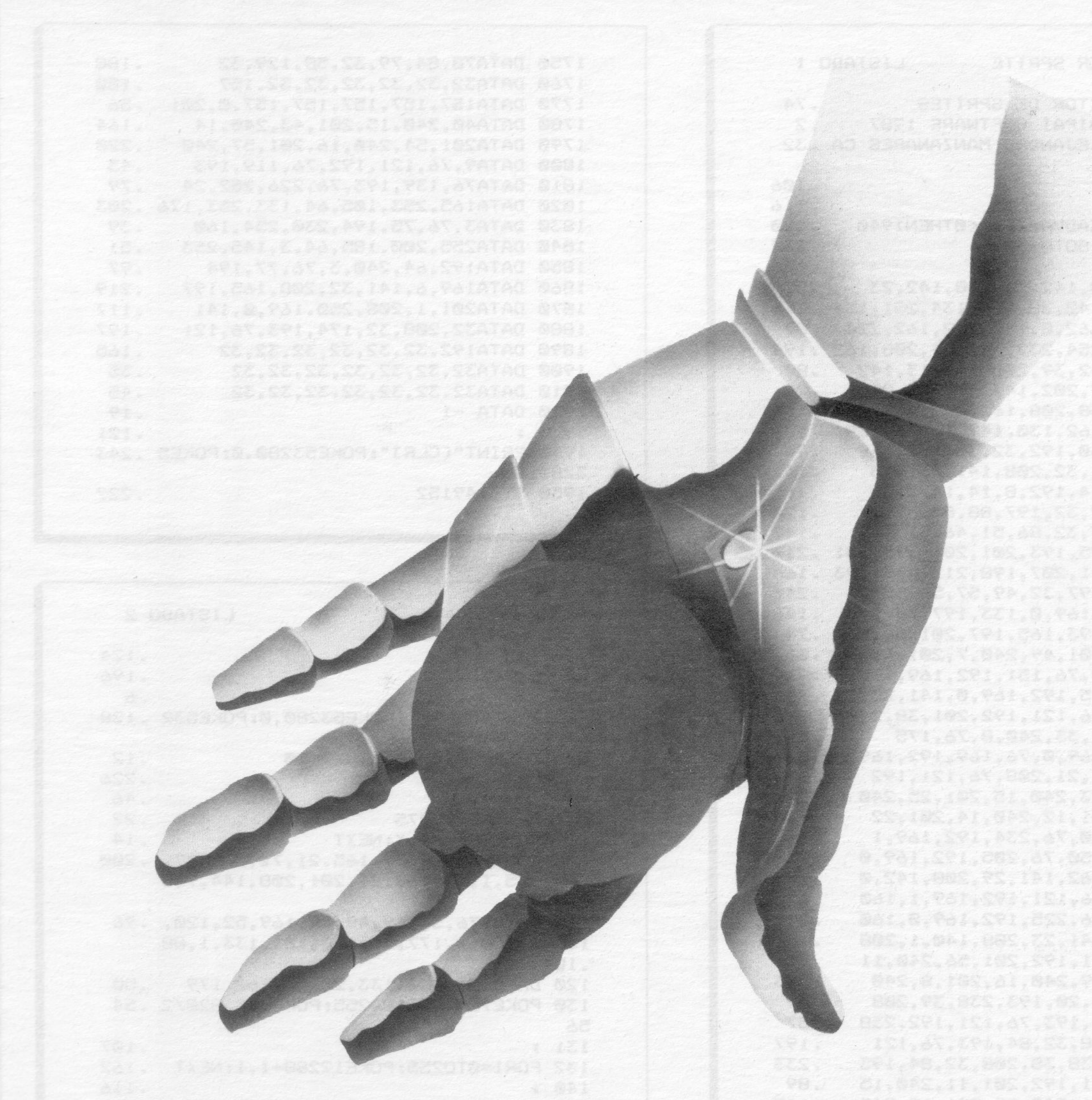
Los controles del programa son los siguientes:

AVANCES:
- [+] — Avance rápido
- [−] — Retroceso rápido
- [S] — Sprite siguiente
- [A] — Sprite anterior

COLORES:
- [1] — Cambia color sprite
- [2] — Cambia multicolor 1
- [3] — Cambia multicolor 2
- [4] — Preset de colores #1
- [5] — Preset de colores #2
- [6] — Preset de colores #3
- [7] — Preset de colores #4
- [*] — Conectar modo multicolor

VARIOS:
- [@] — Salir del modo multicolor
- [X] — Expansión horizontal
- [Z] — «Desexpansión» horizontal
- [Y] — Expansión vertical
- [T] — «Desexpansión» vertical





[I] — Conecta Sprite
[O] — Desconecta Sprite
[F.Izq] — Sale del programa
[F.Arr] — Marca un Sprite para salvarlo

¿Cómo se procede? Primeramente nos movemos por la memoria examinando los diferentes bloques de 64 bytes. Podemos cambiar el color y el modo gráfico, así como expandirlo o no. Si encontramos un Sprite que deseemos extraer del programa que exploramos, lo marcamos pulsando la tecla [F.Arr]. Entonces el borde de la pantalla se pondrá azul, y sólo podremos salir de este estado pulsando la tecla RETURN.

Vamos procediendo de la misma manera hasta que quedamos satisfechos. Entonces salimos del programa (mediante reset o a partir de la opción incluida en el mismo) al BASIC. Cargamos el Generador de Datas (programa DATA) y lo ejecutamos. El programa, tras una breve espera en la que carga una pequeña rutina en Lenguaje Máquina, emite el mensaje «PUEDES PULSAR» y muestra un sprite, que debería ser el primero que hemos marcado para su recuperación. Las opciones del programa se obtienen pulsando [DEL].

Una vez hayamos obtenido las DATAS de un Sprite, éste quedará marcado de nuevo, por lo que su número aparecerá en caracteres inversos.

Finalmente, borramos las líneas en BASIC que hay sobre las DATA de los sprites, y con un simple cargador (que ya se incorpora) pueden utilizarse los sprites en programas propios.

Esto es muy útil para aprender un poco más sobre la estructura de almacenamiento de los gráficos o de como están hechos, que combinaciones de colores son las más espectaculares, etc. Recuerdo a los usuarios que existen programas sobre los que Super Sprite no puede hacer nada ya que almacenan sus sprites de maneras más exóticas. Sin embargo, en un 90% de los casos el programa funciona perfectamente. ■

Para trabajar con sprites, nada mejor que un editor. Y si el editor, además, busca a través de la memoria del ordenador, tanto mejor.

Alejandro Manzanres Cano

```
PROGRAMA: SUPER SPRITE        LISTADO 1

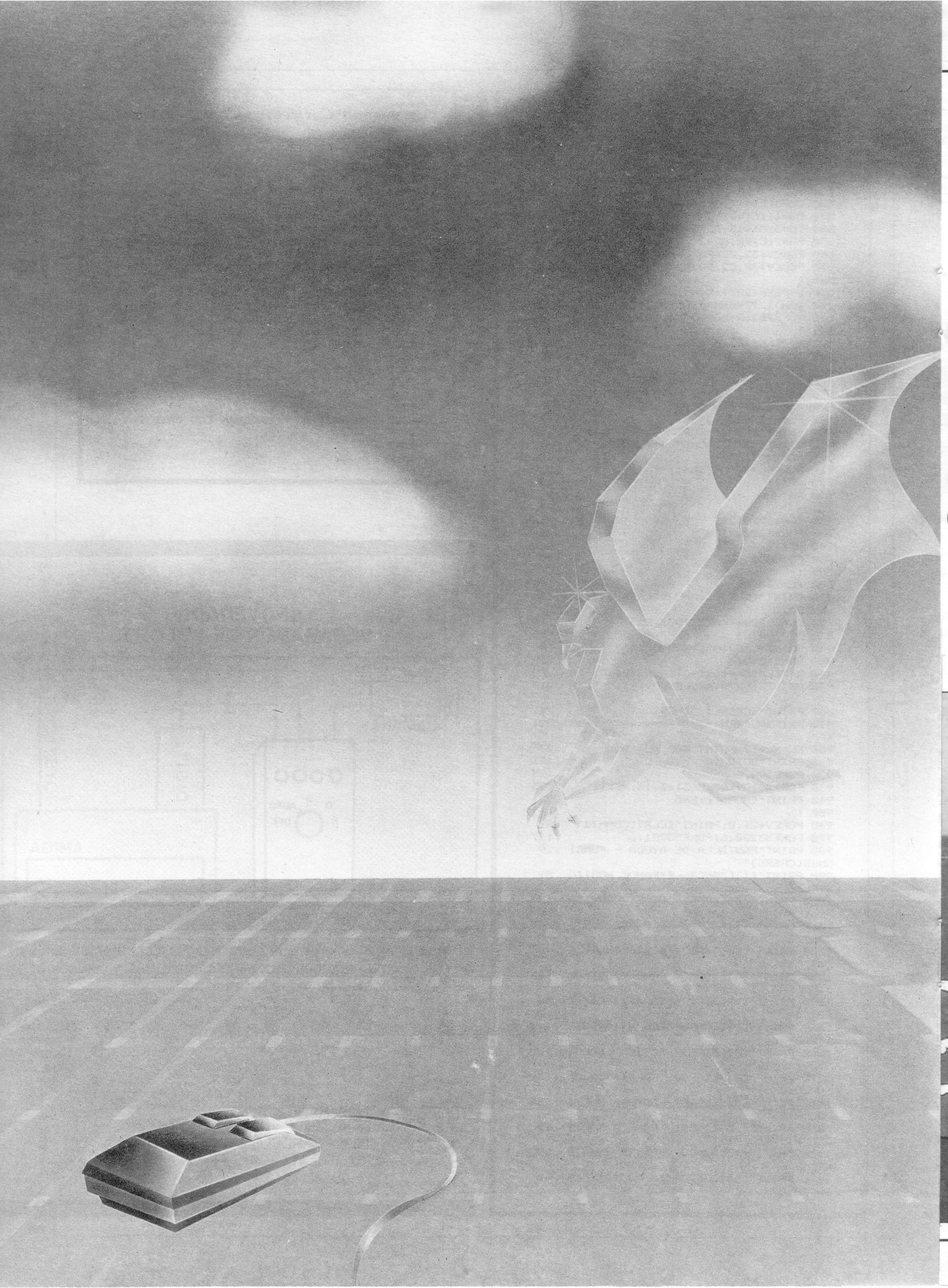
100 REM EXTRACTOR DE SPRITES          .74
110 REM (C) CAIPAI SOFTWARE 1987      .2
120 REM POR ALEJANDRO MANZANARES CA .32
NO
130 :                                 .106
140 OO=49151                          .116
150 OO=OO+1:READVA:IFVA<0THEN1940     .120
160 POKEOO,VA:GOTO150                 .232
170 :                                 .146
1000 DATA162,0,142,29,208,142,23     .85
1010 DATA208,142,28,208,134,251,134 .215
1020 DATA253,162,8,134,252,162.224   .87
1030 DATA134,254,232,142,21,208,162 .194
1040 DATA14.142,39,208,162,13,142    .80
1050 DATA248,7,202,142,37,208,202    .28
1060 DATA142,38,208,162,162,142,0    .150
1070 DATA208,162,130,142,1,208,169   .102
1080 DATA73,160,192,32,30,171,169    .94
1090 DATA0,141,32,208,141,33,208     .142
1100 DATA76,114,192,8,14,147,154     .14
1110 DATA32,32,32,197,88,84,82       .122
1120 DATA65,67,32,86,51,46,51        .148
1130 DATA32,195,193,201,208,193,201 .210
1140 DATA32,211,207,198,212,215,193 .164
1150 DATA210,197,32,49,57,56,55      .240
1160 DATA17,0,169,0,133,197,76       .146
1170 DATA153,193,165,197,201,64,240 .74
1180 DATA250,201,49,240,7,201,46     .86
1190 DATA240,8,76,151,192,169,1      .234
1200 DATA76,145,192,169,0,141,28     .76
1210 DATA208,76,121,192,201,38,240   .176
1220 DATA7,201,33,240,8,76,175       .54
1230 DATA192,169,0,76,169,192,169    .82
1240 DATA1,141,21,208,76,121,192     .112
1250 DATA201,23,240,15,201,25,240    .220
1260 DATA31,201,12,240,14,201,22     .198
1270 DATA240,30,76,234,192,169,1     .50
1280 DATA162,150,76,205,192,169,0    .249
1290 DATA162,162,141,29,208,142,0    .239
1300 DATA208,76,121,192,169,1,160    .173
1310 DATA120,76,225,192,169,0,160    .209
1320 DATA130,141,23,208,140,1,208    .45
1330 DATA76,121,192,201,56,240,11    .69
1340 DATA201,59,240,16,201,8,240     .253
1350 DATA21,76,20,193,238,39,208     .59
1360 DATA32,84,193,76,121,192,238    .57
1370 DATA37,208,32,84,193,76,121     .197
1380 DATA192,238,38,208,32,84,193    .233
1390 DATA76,121,192,201,11,240,15    .89
1400 DATA201,16,240,20,201,19,240    .145
1410 DATA25,201,24,240.30,76,105     .243
1420 DATA193,169,15,162,12,160,11    .171
1430 DATA76,72.193,169,3,162,14      .161
1440 DATA160.6,76,72,193,169,7       .15
1450 DATA162,8,160,2,76,72,193       .125
1460 DATA169,1,162,13,160.5,141      .117
1470 DATA39,208,142,37,208,140,38    .195
1480 DATA208,76,121,192,162,255,160 .95
1490 DATA255,232,224,255,240,3,76    .133
1500 DATA88,193,200,192,60.240,3     .143
1510 DATA76,88,193.96,201,13,240     .77
1520 DATA7,201,10,240,23,76,33       .55
1530 DATA194,32,84,193,24,165,251    .147
1540 DATA105,64,133,251,176,3,76     .182
1550 DATA153,193,230,252,76,153,193 .110
1560 DATA32,84,193,56,165,251,233    .20
1570 DATA64,133,251.144,3,76,153     .202
1580 DATA193,198,252,32,174,193,160 .84
1590 DATA255,200,177,251,153,64,3    .82
1600 DATA192,64,240,3,76,158,193     .192
1610 DATA76,121.192.169,202,160,193 .202
1620 DATA32,30,171,166.251,165,252   .10
1630 DATA32,205,189,169,235,160,193 .112
1640 DATA32,30.171.166.253.165.254   .62
1650 DATA76,205.189,19,30.17,17      .110
1660 DATA17,17,196.73,82,69,67       .236
1670 DATA67,73,79,78,32,32,58        .14
1680 DATA129,32.32,32,32,32,32       .68
1690 DATA32,157,157,157,157,157,157 .6
1700 DATA0,19,156,17,17,17,17        .190
1710 DATA29,29,29,29,29,29,29        .88
1720 DATA29,29,29,29,29,29,29        .98
1730 DATA29,29,29,29,29,29,156       .104
1740 DATA211.65,76.86,65,77,69       .140
1750 DATA78,84,79,32,58.129,32       .188
1760 DATA32,32,32,32,32,32,157       .180
1770 DATA157,157,157,157,157,0,201   .56
1780 DATA40,240,15,201,43,240,14     .164
1790 DATA201,54,240,16,201,57,240    .220
1800 DATA9,76,121,192,76,119,193     .43
1810 DATA76,139,193,76,226,252,24    .79
1820 DATA165,253,105,64,133.253,176 .203
1830 DATA3,76.75.194,230,254,160     .39
1840 DATA255,200,185,64,3,145,253    .51
1850 DATA192.64,240,3,76,77,194      .97
1860 DATA169,6,141,32,208,165,197    .219
1870 DATA201,1,208.250.169,0,141     .117
1880 DATA32,208,32,174,193,76,121    .197
1890 DATA192.32,32,32,32,32,32       .165
1900 DATA32,32,32,32,32,32,32        .35
1910 DATA32.32,32,32,32,32,32        .45
1920 DATA -1                          .19
1930 :                                .121
1940 PRINT"[CLR]":POKE53280,0:POKE5 .243
3281.0
1950 SYS 49152                        .227
```

```
PROGRAMA: DATA                     LISTADO 2

10 CLR                                .124
20 POKE650.128                        .196
30 :                                  .6
40 PRINT"[CLR]":POKE53280,0:POKE532 .128
81,0
50 IFPEEK(828)=165THEN150             .12
60 POKE787,0                          .226
70 :                                  .46
80 FORI=828TO875                      .22
90 READX:POKEI.X:NEXT                 .14
100 DATA165,20,72,165,21,72,32,247, .200
183,165,1,72,165,21,201,208,144,7,2
01,224
110 DATA176,3,169,49.44,169,52,120, .96
133,1,160,0,177,20,168.104,133,1,88
.104,133
120 DATA 21,104.133,20.76,162,179    .50
130 POKE785,828AND255:POKE786,828/2 .54
56
131 :                                 .107
132 FORI=0TO255:POKE12288+I,1:NEXT   .162
140 :                                 .116
150 V=53248:POKEV,159:POKEV+1,190:P .148
OKEV+39.12:POKEV+38,11:POKEV+37,15:
POKE787,1
160 POKE2040,15:OO=57343:BA=959      .228
170 POKEV+21,1:CO=12:C1=15:C2=11     .142
180 PRINT"[HOM][GRN][16CRSRD][13SHI .120
FT*]PUEDES[SHIFT*]PULSAR[14SHIFT*][
COMM1]"
190 :                                 .166
200 SM=PEEK(197):IFSM=64THEN200       .108
210 :                                 .186
220 IFSM=18THEN800                    .144
230 IFSM=23THENPOKEV+29,1             .118
240 IFSM=12THENPOKEV+29,0             .64
250 IFSM=25THENPOKEV+23,1             .200
260 IFSM=22THENPOKEV+23,0             .211
270 IFSM=46THENPOKEV+28.1             .131
280 IFSM=49THENPOKEV+28,0             .141
290 IFSM=38THENPOKEV+21,0             .243
300 IFSM=41THENPOKEV+21.1             .59
310 IFSM=4THENGOTO450                 .83
320 IFSM=5THEN470                     .183
330 IFSM=6THEN490                     .5
340 IFSM=56THEN520                    .13
350 IFSM=59THEN530                    .155
360 IFSM=8THEN540                     .155
370 IFSM=11THEN550                    .177
380 IFSM=40THENGOSUB590               .251
390 IFSM=43THENGOSUB630               .91
400 IFSM=57THENPOKEV+21,0:PRINT"[CL .105
R]":LIST10000-
410 IFSM=60THEN750                    .159
420 IFSM=0THENGOSUB950:GOTO180        .59
430 GOTO200                           .113
440 :                                 .161
```

```
450 CO=CO+1:IFCO>15THENCO=CO-15         .223
460 POKEV+39,CO:GOTO200                 .157
470 C1=C1+1:IFC1>15THENC1=C1-15         .93
480 POKEV+37,C1:GOTO200                 .245
490 C2=C2+1:IFC2>15THENC2=C2-15         .237
500 POKEV+38,C2:GOTO200                 .75
510 :                                   .231
520 CO=3:C1=14:C2=6:GOTO560             .40
530 CO=11:C1=12:C2=15:GOTO560           .70
540 CO=5:C1=13:C2=1:GOTO560             .112
550 CO=7:C1=8:C2=2                      .124
560 POKEV+39,CO:POKEV+37,C1             .214
570 POKEV+38,C2:GOTO200                 .146
580 :                                   .46
590 N=PEEK(787):N=N+1                   .88
600 IFN>127THENN=1:OO=57343             .254
610 POKE787,N:OO=OO+64:GOTO670          .100
620 :                                   .86
630 N=PEEK(787):N=N-1                   .192
640 IFN<1THENN=127:OO=65472             .54
650 POKE787,N:OO=OO-64                  .6
660 :                                   .126
670 IFPEEK(12288+PEEK(787))=255THEN     .28
FL$="[RVSON]":GOTO690
680 FL$="[RVSOFF]"                      .178
690 PRINT"[HOM][18CRSRD][4CRSRR]"FL     .108
$"SPRITE:[4SPC][4CRSRL]"PEEK(787)
700 FORI=0TO63                          .186
710 POKEBA+I,USR(OO+I)                  .232
720 NEXTI                               .38
730 RETURN                              .22
740 :                                   .206
750 FORI=0TO128                         .214
760 POKE12288+I,0                       .228
770 NEXTI                               .89
780 GOTO430                             .17
790 :                                   .1
800 NL= 10000                           .197
810 PRINT"[CLR][COMM5][2CRSRD]";:PO     .103
KEV+21,0
820 POKE12288+PEEK(787),255             .119
830 FORI=0TO3                           .135
840 PRINTNL"DATA";:NL=NL+10             .15
850 FORJ=0TO15                          .215
860 A=PEEK(BA+16*I+J):A$=STR$(A)        .207
870 PRINTRIGHT$(A$,LEN(A$)-1)",";:N     .219
EXT
880 PRINT"[CRSRL] ":NEXT                .241
890 PRINTNL"REM ===================     .221
=="
900 NL=NL+10:PRINT"800 NL="NL           .239
910 PRINT"GOTO 10"                      .121
920 POKE198,9                           .199
930 FORI=0TO9:POKE631+I,13:NEXTI        .195
940 PRINT"[HOM]";:END                   .9
950 :                                   .161
960 POKEV+21,0:PRINT"[CLR][COMM7]"      .87
970 POKE53280,6:POKE53281,6             .117
980 PRINT"PANTALLA DE AYUDA - FUNCI     .19
ONES[CRSRD]"
990 PRINT"[X][3SPC]- EXPANDE HORIZO     .59
NTAL"
1000 PRINT"[Z][3SPC]- DESEXPANDE[2S     .183
PC]''"
1010 PRINT"[Y][3SPC]- EXPANDE VERTI     .245
CAL"
1020 PRINT"[T][3SPC]- DESEXPANDE[2S     .9
PC]''"
1030 PRINT"[@][3SPC]- PONE MODO MUL     .198
TICOLOR"
1040 PRINT"[*][3SPC]- SACA[2SPC]''[     .38
4SPC]''"
1050 PRINT"[O][3SPC]- DESACTIVA SPR     .110
ITE"
1060 PRINT"[P][3SPC]- ACTIVA[6SPC]'     .138
'"
1070 PRINT"[F1][2SPC]- CAMBIA COLOR     .134
 PROPIO"
1080 PRINT"[F3][2SPC]- CAMBIA MULTI     .250
COLOR 1"
1090 PRINT"[F5][2SPC]- CAMBIA MULTI     .198
COLOR 2"
1100 PRINT"[+][3SPC]- SPRITE SIGUIE     .200
NTE"
1110 PRINT"[-][3SPC]-[3SPC]''[3SPC]     .40
ANTERIOR"
```

```
1120 PRINT"[_][3SPC]- SALE DEL PROG     .44
RAMA"
1130 PRINT"[1][3SPC]- JUEGO DE TONO     .250
S 1"
1140 PRINT"[2][3SPC]-[3SPC]''[6SHIF     .204
T SPC]''[3SHIFT SPC]2"
1150 PRINT"[3][3SPC]-[3SPC]''[6SHIF     .254
T SPC]''[3SHIFT SPC]3"
1160 PRINT"[4][3SPC]-[3SPC]''[6SHIF     .48
T SPC]''[3SHIFT SPC]4"
1170 PRINT"[ ][3SPC]- INICIALIZA SI     .80
STEMA"
1180 PRINT"[D][3SPC]- CREA SENTENCI     .244
AS DATA
1190 PRINT"[CRSRD]PULSA UNA TECLA"      .34
1200 IFPEEK(197)=64THEN1200             .8
1210 PRINT"[CLR]":POKEV+21,1            .244
1220 POKE53280,0:POKE53281,0            .76
1230 RETURN                             .12
1240 :                                  .196
1250 OPEN15,8,15,"S0:EXTRAC.DATA":C     .146
LOSE15:SAVE"EXTRAC.DATA",8
1260 :                                  .216
1290 :                                  .247
1300 REM AQUI SE ALMACENAN LOS DATO     .47
S
1310 :                                  .11
1320 :                                  .21
```





COLORES POR DEFECTO PARA EL C-128

Alguno de los ingeniosos creadores del Commodore 128 debió de tener un extraño sentido del humor cuando seleccionó la combinación de colores verde/gris sopa como colores por defecto para la pantalla de 40 columnas del C-128. Pero lo peor es que no importa si cambias de colores mediante POKEs, cada vez que pulsas STOP/RESTORE, vuelven a aparecer los verdes y grises en la pantalla.

Mi programa, Colores por Defecto C-128, instala la combinación de colores de azul claro/oscuro como colores de los caracteres y del fondo de la pantalla, los mismos que en el C-64. Creo que resulta mucho más agradable a la vista. Aun cuando pulses STOP/RESTORE, seguirás con los mismos colores por defecto, no los de Commodore. Para seleccionar tus propios colores, cambia los dos números 14 (colores del borde y del texto) y el 6 (color de la pantalla) en las instrucciones DATA a los valores deseados.

```
PROGRAMA: COLORES                    LISTADO 1

0 REM COLORES DE C-64                     .126
10 FORA=3072TO3099:READD:CK=CK+D:PO       .66
KEA,D:NEXT
20 IFCK<>2581THENPRINT"ERROR EN DAT       .158
AS...":END
30 SYS3072                                .168
40 DATA 169,11,141,0,10,169,12,141,       .244
1,10,96,169,14,141,32,208,169
50 DATA 6,141,33,208,169,14,133,241       .68
,76,3,64
```

Bo Cochran

CONTROL-G EN EL C-64

La mayoría de los propietarios de un C-128 saben que al pulsar la secuencia CTRL-G en sus ordenadores hace sonar una campanilla a través del altavoz del monitor. Esta pequeña rutina para C-64 proporciona al «hermano pequeño» una combinación de teclas que suena de forma idéntica.

```
PROGRAMA: CAMPANILLA                 LISTADO 2

0 REM CAMPANILLA C-64                     .86
10 PRINT"[CLR]PULSA CTRL-G PARA OIR       .112
LA CAMPANILLA"
20 GETK$:IFK$<>"[CTRL G]"THEN20           .82
30 GOSUB50                                .2
40 GOTO20                                 .228
50 V=54273:POKEV+23,21:POKEV+4,9:PO       .230
KEV+5,0:POKEV,48:POKEV+3,32:POKEV+3
,33
60 RETURN                                 .118
```

Barbara Schulak

CARACTERES DESDE EL CENTRO

Hacer que los caracteres «crezcan» hacia los lados desde el centro de la pantalla es un truco muy vistoso. Extractor de Letras, el programa que tienes listado a continuación funciona en los modos 64 y 128. Sólo tienes que asignar un mensaje a la cadena A$ y podrás utilizarla repetidas veces desde tus programas con el comando GOSUB 50000.

```
PROGRAMA: EXTRACTOR                  LISTADO 3

0 REM EXTRACTOR DE LETRAS                 .162
10 PRINTCHR$(147):PRINT"[WHT]"            .170
20 A$="PULSA UNA TECLA PARA OTRO ME       .208
NSAJE!":GOSUB50000
25 GETA$:IFA$=""THEN25                    .231
30 A$="[5SPC]ESTE ES EL SEGUNDO MEN       .210
SAJE![4SPC]":GOSUB50000
40 PRINT:END                              .104
50000 A=LEN(A$):IFA/2<>INT(A/2)THEN       .145
A$=A$+"[RVSON]"
50010 FORI=1TOA/2:PRINT"[HOM]"TAB(2       .31
2-I);LEFT$(A$,I)RIGHT$(A$,I):PRINT"
[2CRSRD]"
50015 FORH=1TO40:NEXT:NEXT:RETURN         .148
```

Brian K. Jandula

BLOQUES NECESARIOS

Si eres de los que habitualmente llenas todo el espacio de tus discos, prueba a introducir la siguiente línea en tu ordenador. Te dirá cuántos bloques necesitas antes de salvar un programa a disco.

Para el C-64:

X=PEEK(45)+256*PEEK(46)-2049:PRINT INT(X/254)+1;«BLOQUES»

Para el C-128:

GRAPHIC CLR:X=58109-FRE(0):PRINT INT(X/254)+1;«BLOQUES»

En el caso del C-64, el programa calcula el final del programa Basic y le resta la dirección inicial (2049). En la versión para 128 esto es más fácil de calcular, porque todo el banco 0 excepto 7427 bytes están disponibles para el almacenamiento de programas.

Jeff Ross

BORRADO LINEA A LINEA EN EL C-64

Cualquiera que escriba programas para el C-64 habrá sentido alguna vez la necesidad de borrar parte de la pantalla, pero dejando el resto intacto. Hay varios procedimientos para hacerlo, como mantener pulsada la barra de espacios o la tecla de borrar.

Borrador de Líneas C-64 proporciona un sistema mucho más rápido. Sólo tienes que teclear este programa, grabarlo y a continuación ejecutarlo. Mientras esté activado, no podrás saber si está presente hasta que pulses la tecla de borrar. Para hacerlo, pulsa la tecla F1 y... ¡zap! ya no hay línea. El programa desde el que borras la línea permanece sin modificación, y la línea todavía queda en la memoria, para el caso de que quieras volver a dejarlo todo como estaba.

```
PROGRAMA: BORRADOR                 LISTADO 4

0 REM BORRADOR DE LINEAS C-64          .62
1 REM (C)1989 BY MARC TEMANSON         .219
2 REM (C)1989 BY COMMODORE WORLD       .18
3 :                                    .235
10 FORX=49152TO49193:READS:POKEX,S:    .32
NEXT:SYS49152
20 DATA 120,169,192,141,21,3,169,13    .192
,141,20,3,88,96,165,207,201,0,240
30 DATA 3,76,49,234,165,203,201,4,2    .242
40,3,76,49,234,166,214,142,13,3
40 DATA 32,255,233,76,49,234           .166
```

FUNDIDOS C-64

Algunos programas comerciales utilizan una técnica llamada «fundido» que hace que las imágenes aparezcan y desaparezcan lentamente de la pantalla. Hasta ahora, utilizar esta técnica en los programas de uno mismo era algo que quedaba lejos del alcance de todos excepto algunos expertos programadores.

Ahora, utilizando el C-64 y su juego de caracteres programables, junto con este pequeño programa que tienes a continuación, cualquiera puede hacerlo. Teclea el programa, utilizando el programa comprobador de sumas de control PERFECTO, y grábalo. A continuación, ejecuta el programa y observa cómo la pantalla rivaliza con los efectos especiales de muchos programas comerciales.

Cuando introduces por primera vez el comando SYS 49152, el contenido de la pantalla desaparece, y la segunda vez vuelve a aparecer. Para variar la velocidad, teclea POKE 49240, seguido de una coma y de un número entre 0 (el más rápido) y 255 (el más lento). Incluyendo esta técnica en tus propios programas puedes crear presentaciones espectaculares.

```
PROGRAMA: FUNDIDOS                 LISTADO 5

0 REM FUNDIDOS-64                      .202
1 REM (C)1989 BY JEFF A. MARTIN        .75
2 REM (C)1989 BY COMMODORE WORLD       .18
3 :                                    .235
10 FORI=49152TO49420:READD:POKEI,D:    .110
C=C+D:NEXTI
20 IFC<>31597THENPRINT"ERROR EN DAT    .222
AS...":END
30 FORT=1TO20:PRINT" LETTER FADER I    .94
S ACTIVATED":SYS49152:NEXT
40 DATA 173,24,208,41,8,208,62,32,2    .90
7,192,208,40,160,0,132,250,132,252
50 DATA 169,208,133,253,169,48,133,    .138
251,96,32,12,192,133,52,133,56,173
60 DATA 24,208,41,240,9,12,141,24,2    .0
08,140,14,220,169,51,133,1,96,177
70 DATA 252,145,250,200,208,249,230    .242
,251,230,253,165,253,201,224,208,23
9
80 DATA 32,44,192,32,12,192,160,0,1    .46
69,192,133,255,238,11,193,173,11,19
3
```

```
90 DATA 201,1,208,246,169,255,141,1 .54
1,193,166,255,169,248,133,254,173,1
2
100 DATA 193,240,5,32,171,192,80,3, .92
32,187,192,232,200,240,6,230,254,20
8
110 DATA 235,240,212,230,251,230,25 .230
3,165,253,201,210,208,240,32,12,192
120 DATA 24,165,255,105,8,133,255,2 .36
08,190,173,12,193,240,5,142,12,193
130 DATA 208,4,232,142,12,193,169,5 .212
5,133,1,169,1,141,14,220,96,177,252
140 DATA 61,11,192,141,13,193,177,2 .70
50,13,13,193,145,250,96,189,11,192,
73
150 DATA 255,141,13,193,177,250,45, .72
13,193,145,250,96,16,4,4,1,1,64,128
160 DATA 32,64,64,64,2,4,8,64,1,128 .182
,32,128,16,128,16,8,128,4,128,32,64
170 DATA 32,128,1,2,8,8,8,128,64,32 .152
,2,8,32,1,1,32,16,1,4,16,1,16,2,8,2
,2
180 DATA 32,4,2,2,16,4,8,4,16,64,25 .86
5,0
```

TURBO CURSOR

Estos comandos POKE hacen que el cursor se mueva mucho más deprisa de lo normal a través de la pantalla, lo cual resulta muy útil e interesante para los programas que aprovechan mucho la presentación en pantalla. Sólo tienes que introducir esa línea:

POKE 650,128:POKE 56325,10

Al hacer un POKE en la posición 650 con el valor 128 se consigue que todas las teclas sean repetitivas. Puedes cambiar la velocidad del cursor haciendo POKEs con diferentes valores en la posición 56325 (los números más bajos proporcionan velocidades más altas). ¿Puede alguien escribir una rutina para mover el cursor diagonalmente en todas direcciones?

M. Pellegrino

TECLADO MORSE 128

Este pequeña rutina permite convertir la tecla SHIFT del C-128 en un «transmisor morse». Puede utilizarse como rutina en tus propios programas, o como programa autónomo. Puedes modificar el tono del sonido cambiando el valor 3000 de la línea 20.

```
PROGRAMA: MORSE 128                LISTADO 6

0 REM MORSE-128                         .118
1 REM (C)1989 BY LOUIS R. FERNANDEZ .91
2 REM (C)1989 BY COMMODORE WORLD    .18
3 :                                     .235
10 IFPEEK(211)=0THEN10                  .162
20 SOUND1,1E4,3000                      .54
30 IFPEEK(211)=1THEN30                  .218
40 SOUND1,1,0                           .42
50 GOTO10                               .230
```

Louis R. Fernández

Sección de

JUEGOS

AMIGA

AFRICAN RAIDERS

222

El rally París-Dakar ha inspirado numerosos juegos como el que vamos a comentar, pero son escasos los que se han acercado a la realidad de la prueba. African Raiders es una aventura a través del desierto africano, pilotando un rapidísimo 4x4.

La carrera comienza en Túnez y termina, tras cinco durísimas etapas, en la costa de Dakar. A lo largo de esas etapas se recorren las dunas del desierto, las zonas montañosas, las zonas de nómadas donde te tropiezas con un montón de camellos, etc. Al arrancar tienes a tu lado diversos vehículos que también participan en la competición. Ellos van muy de prisa y nunca se pierden, porque no se salen del camino marcado por los bidones. Pero tú debes cuidar la conducción y afinar en las trazadas de las curvas para no perderte en el desierto. El tema de perderse en el desierto, aunque parezca malo, tiene su encanto, ya que permite aprender a orientarse y volver a descubrir por uno mismo el camino correcto. Este es el único juego conocido que permita perderse por enormes espacios fuera del camino correcto.

La acción es trepidante. El coche puede rodar con tracción normal o a las cuatro ruedas. Cuando saltas sobre una duna, la impresión es muy real y divertida. Al derrapar salta la arena y cuando chocas con un objeto contundente puedes llegar a volcar. Se debe tener cuidado ya que los instrumentos del tablero se van deteriorando con los golpes. Y estos instrumentos son muy importantes para no perderse. Los que aparecen en el tablero son: El indicador de la caja de cambios (marcha adelante, punto muerto y marcha atrás), indicador de tracción (2 ó 4 ruedas motrices), el inevitable indicador de velocidad, el contador kilométrico de la etapa, una útil brújula electrónica y el indicador del carburante.

Los gráficos del juego son realmente muy buenos. Después de llevar varias horas jugando, se empieza a notar la similitud del paisaje y de los elementos del terreno, pero tampoco es el punto más importante del juego. Por otra parte, los árabes que encuentras junto a algún barril o los indicadores que se muestran en ciertos puntos el camino, están bien dibujados. También aparecen camellos, restos de chatarra de coches, etc. El gráfico más importante, el del coche, está muy bien dibujado. En realidad son varios sprites los que recrean la imagen del coche en movimiento, ya que los giros son variados, así como los derrapajes.



El sonido combina muy bien con una animación más acertada todavía. No es un programa que se preste a introducir músicas, pero los efectos sonoros son buenos. El mejor consejo en el rally es ser prudente. No te salgas nunca de la línea marcada por los bidones y menos sin copiloto. Cuando tengas más experiencia y si consigues jugar junto a un amigo que controle bien el mapa del desierto, puedes aventurarte atravesándolo para acortar terreno y llegar antes a la meta. En cualquier caso, buena suerte.

Por último, para quienes consideren este juego demasiado difícil, animarles a que vengan despacito, aprendiendo a disputar de un 4×4.

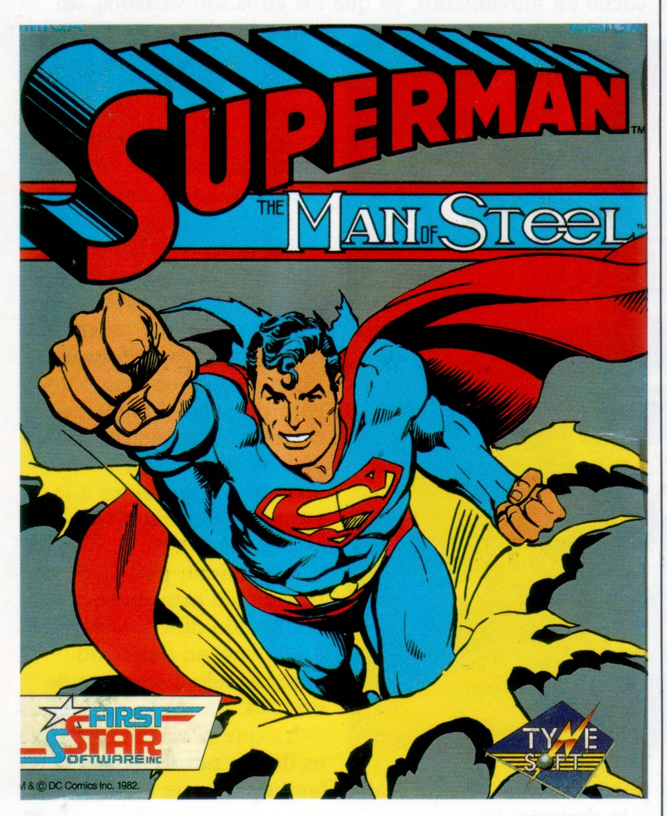
Ya que el cartel estricto es difícil, por lo menos se puede entretener uno deambulando por entre los camellos y la chatarra. ■

En el próximo número, además de otros juegos, comentaremos:
JUANA DE ARCO,
THE LORDS OF THE RISING SUN, y
CRAZY CARS II

SUPERMAN

223

El legendario Superman, aparece en las pantallas del Amiga de la manera más espectacular. La gran dificultad, aparece cuando uno se da cuenta, que los enemigos de Superman en este juego para ordenador, no son tan fáciles de derrotar como los que aparecen en las películas. Precisamente, es la dificultad uno de sus atractivos. Los gráficos y sobre todo la perfecta adaptación al ordenador de la famosa banda sonora de la película, conforman este programa de gran calidad. Al componerse



de dos discos, y ser los gráficos tan buenos, el programa acude a menudo al disco, lo que da una pequeña sensación de lentitud.

Es un poco pesado, sobre todo hasta que se coge el truquillo, el inicio del juego, ya que, aunque se dispone de tres vidas, cada vez que se pierde una se vuelve al inicio. Esto supone nueva grabación, presentación etc, mientras que todos desearíamos un rápido inicio para no perder el calor de la lucha.

La introducción al juego es sencilla, Superman se encuentra en la redacción del Daily Planet (sin que nadie sepa su verdadera personalidad claro), y son avisados de que unos terribles invasores alienígenas amenazan la tierra. De inmediato Superman se transforma y vuela hacia su encuentro. Es en este punto donde empiezan las dificultades.

Después de una lenta grabación de disco, empieza el juego. Superman se enfrenta con unos invasores procedentes de un sol negro en extinción. La fuerza de Superman está marcada por una línea de cuadritos en la parte superior derecha. Los invasores que al igual que Superman también pueden volar, atacan lanzando bolas de fuego que, aunque no matan, a Superman sí le hacen perder fuerza. Las pequeñas naves invasoras pilotadas por un alienígena, son aún más peligrosas ya que éstas lanzan un campo magnético que se va ampliando según va acercándose. Para destruir a tantos enemigos, se dispone de tres armas: la vista calorífica (rayos que Superman dispara con los ojos), su superpuño y por último el viento que lanza cuando sopla.

Para eliminar a los enemigos, la forma de hacerlo es principalmente con la vista y el puño. Disparando continuamente contra una nave conseguirás derribarla aunque es más fácil hacerlo de una sola vez con el puño, si se da a la nave, ésta saldrá disparada hacia atrás pero no será destruida, si se da al piloto de la nave ésta caerá sin control. La destrucción total de los invasores se hace realmente difícil, por no decir imposible, teniendo en cuenta que continuamente llegan más y su destrucción no es tan fácil. Cuando Superman tiene poca energía, se recupera con tiempo, por tanto y con el fin de ganarlo, lo mejor es «soplar», con lo cual alejaremos a los alienígenas durante algún tiempo.

El descubrimiento de las siguientes pantallas es para los muy adiestrados en la lucha... A los ya indicados poderes de Superman hay que sumar la posibilidad de volar que en la primera parte se da por supuesta y no se agota. En las posteriores operaciones de Superman, la posibilidad de volar no es eterna y como en la primera parte, de utilizarlo mucho se agota y hay que descansar sin utilizarlo para que nos permita volver ha hacerlo. Las instrucciones del programa dejan mucho que desear ya que no desvelan ninguna de las posibilidades con que se puede derrotar a los enemigos por lo que no sirven para mucho. Superman, un clásico del cómic, ha sido llevado con acierto a las pantallas de los ordenadores, a pesar de la dificultad. ■

Sección de JUEGOS

AMIGA

DOUBLE DRAGON Y ALIEN SYNDROME 224

DOUBLE DRAGON

Seguramente todos conocéis este juego por las máquinas de los salones recreativos, en ellas es realmente famoso. En el Commodore, lógicamente, no es tan bueno, pero tanto la acción como los gráficos etc, están bastante bien.

La acción del juego se desarrolla en las calles de una gran ciudad, en la que como en cualquiera, hay riñas callejeras entre bandas etc. Los hermanos gemelos Billy Lee y Jimmy Lee son de los pocos que se atreven con estas bandas y por ello la más peligrosa banda ha secuestrado a la novia de Billy. Para rescatarla le ayudará su hermano, entre los dos se tendrán que enfrentar con todos los enemigos de la calle. Después de haber recorrido las calles, llegarán a la parte industrial donde tendrán que subirse por cajas etc almacenas para así poder seguir avanzado. Seguidamente pasarán a un parque y por fin a la guarida donde está el jefe de la banda Willy, con una altura de 1,82 m. y un peso de 92 Kg. lucha con gran maestría y está armado con una ametralladora que no dudará en usar.

Los gráficos de fondo son sencillos y repetitivos pero por el contrario los movimientos de los luchadores son buenos. El juego no pierde en velocidad al jugar dos, que es como más se aprovechan las cualidades de éste. También se puede jugar uno y unirse el otro jugador en los momentos más difíciles.

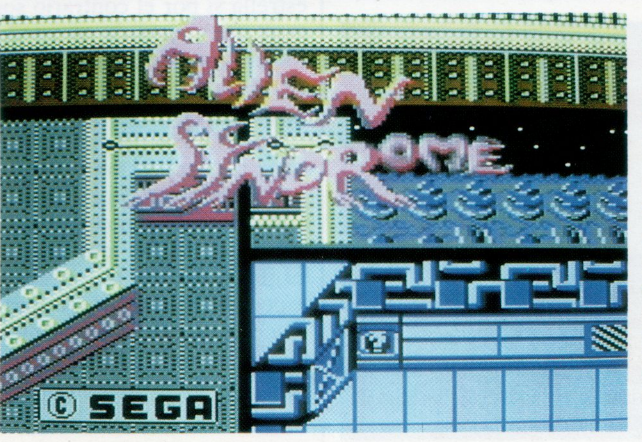
En la parte inferior de la pantalla se indica la fuerza de que dispone cada jugador así como las vidas que le quedan. Como en las máquinas de los salones, se tiene la oportunidad de introducir otra moneda y seguir jugando en la misma parte del juego en que se perdió la última vida. Para poderlo hacer sólo se dispone de cinco CREDITOS que también están marcados en la parte inferior de la pantalla.

ALIEN SYNDROME

Alien Syndrome, se parece un poco al desarrollo de la conocida película Alien el regreso. En el juego, al igual que en la película, también hay que intentar rescatar a los humanos atrapados. En la parte superior de la pantalla están representados con pequeñas cabezas los compañeros que aún queden por rescatar, su situación en la base, puede ser detectada gracias a los Mapas de situación que encontrarás muy a menudo. Al igual que los mapas, están repartidas armas especiales que harán más fácil la destrucción de los monstruos que continuamente atacan.

Continuamente durante la misión, se lucha contra el tiempo ya que una bomba oculta en la base tiene un dispositivo especial de cuenta atrás que no admite ninguna posibilidad de ser anulado. El contador del tiempo que queda está en la pantalla. También serás informado mediante apremiantes mensajes para que te des prisa en lograr rescatar a todos los compañeros. Una vez logrado, en la parte superior se encuentra la salida. Al pasar a la siguiente zona te encontrarás con el monstruo que continuamente intentará destruirte para matarlo, hay que disponer de las mejores armas que puedas coger en la parte anterior.





Este juego tiene la posibilidad de los dos jugadores con lo que se hace un poco más fácil siempre que se esté lo suficientemente compenetrado con el compañero. Al jugar dos, no se pierde velocidad y sí resulta más divertido.
Estos dos programas, ya comentados para Amiga, son de gran calidad en la versión Commodore 64, ■

Sección de JUEGOS

THE MUNCHER

225

Tu eres un gran monstruo, y tu objetivo en este juego es atravesar Japón en busca de unos huevos robados por exploradores japoneses. Los exploradores no se dieron cuenta, que al hacerlo despertaron a un terrible monstruo prehistórico que es quién irá a recuperarlos. Este monstruo es THE MUNCHER, dotado de una gran altura con la que alcanzará los altos edificios del Japón y también a los helicópteros que intentarán destruirte.

Inicialmente aparecerás en el principio de una gran ciudad, desde tu aparición el ejército japonés se prepara para destruirte, continuamente aparecerán soldados entre la población civil que dispararán continuamente contra The Muncher. Puedes comerte a los soldados, o también, aplastarlos de un coletazo. Entre los soldados se acercarán camiones y jeeps militares que también pueden ser destruidos de la misma forma.





Muncher tiene bastante fuerza como para destrozar a unos cuantos enemigos, pero ésta se irá acabando dependiendo de los disparos que te acierten los soldados, helicópteros etc, que con potentes armas dispararán continuamente. Es por eso que no te dará tiempo a destruirlos a todos antes de que consigan disparar contra ti. Por ello que cuando puedas, tendrás que reponer fuerzas. Las fuerzas se reponen en parte, comiendo residuos radiactivos en las centrales, pero éstas han de servir principalmente pera incubar los huevos robados que irás encontrando paulatinamente.

Las centrales nucleares, son uno de los lugares más importantes en el juego, cada huevo que en ellas se ponga a incubar, es como una vida más que tendrás. Una vez destruido el primer monstruo, serán sus hijos los que acaben la misión, por tanto en la última central donde dejaste a incubar un huevo, será donde reaparezcas y desde allí reiniciar la misión. Si la salud del monstruo no es buena y no está muy lejos el último sitio en que incubaste un huevo, lo mejor es comerse la siguiente central, de esta manera evitarás ser destruido y no perderás la oportunidad de poder reaparecer en ese lugar.

Los controles del juego son bastante delicados y un poco complicados en cuanto a los movimientos del monstruo se refiere. El movimiento cambia dependiendo de si el disparo está o no pulsado, también cambia cuando The Muncher está subiendo por un edificio.

Una de las cualidades mejores del monstruo, es la posibilidad de disparar fuego por la boca. En un principio tiene cuatro bolas de fuego, pero éstas pueden aumentar si comes un camión de gasolina, pero hay que tener mucho cuidado, los camiones de gasolina están marcados con una estrella si por el contrario son azules, son camiones de agua que dejarán a cero todas las reservas de bolas de fuego acumuladas. Cada camión de gasolina proporciona cuatro bolas de fuego que se sumarán a las que ya dispongas.

Los gráficos del juego dejan mucho que desear, pero el movimiento y la música están bien. El juego resulta un poco repetitivo pero con la posibilidad de subirse a los edificios y la gran cantidad de movimientos diferentes del monstruo resulta más entretenido.
El juego no resulta espectacular a nivel de sonido (efectos especiales y música). A pesar de lo cual es divertido y con un poco de imaginación por parte del jugador, se pueden alcanzar grandes puntuaciones. Es un juego en el que se puede durar lo suficiente como para cogerle gustillo. ■

Sección de JUEGOS

CUATRO X UNO

226

La oferta de cuatro por uno es en este caso bastante interesante. Una de las curiosidades de este tipo de juegos es que, durante la carga de los programas, te permite jugar un poco a un sencillo programa de «marcianitos». En la primera cara se encuentran los juegos SCUMBALL y DROIDS, en la segunda BLACKWYCHE y BOSCONIAN 87.

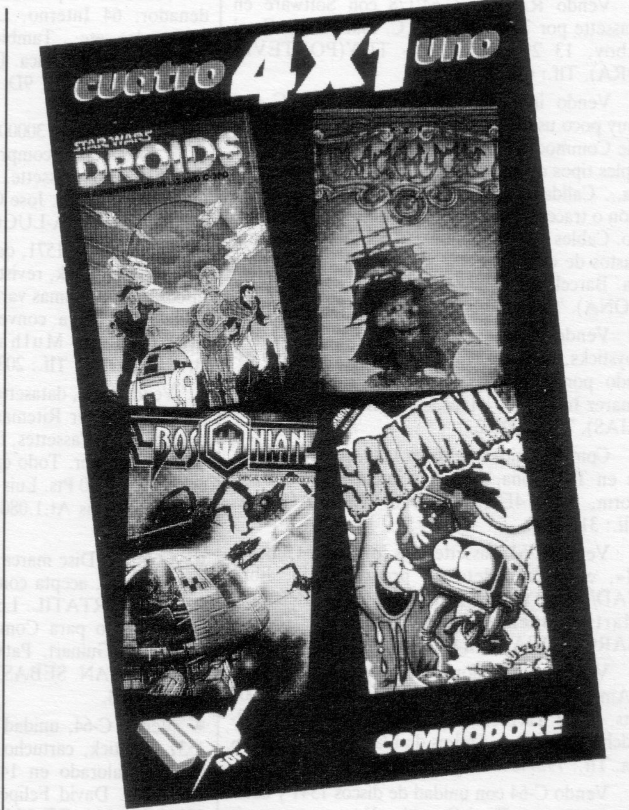
SCUMBALL, es un divertido juego en el que se maneja un robot por las sucias cloacas de una gran ciudad del futuro. En ellas encontrarás ocho granadas que hay que recoger. Cada vez que se coge una granada, hay que trasladarla a la guarida del Gran Fango Verde. Cuando estén allí las ocho granadas, será destruido el Gran Fango. Por todas las cloacas se encuentran alienígenas que intentarán la destrucción del robot. Cada vez que sea tocado perderá fuerza y una vez perdida toda, perderá la vida. Los alienígenas etc, pueden ser destruidos disparando sobre ellos. Si en el recorrido caes sobre el agua o en los fosos con clavos, perderás la vida en el acto.

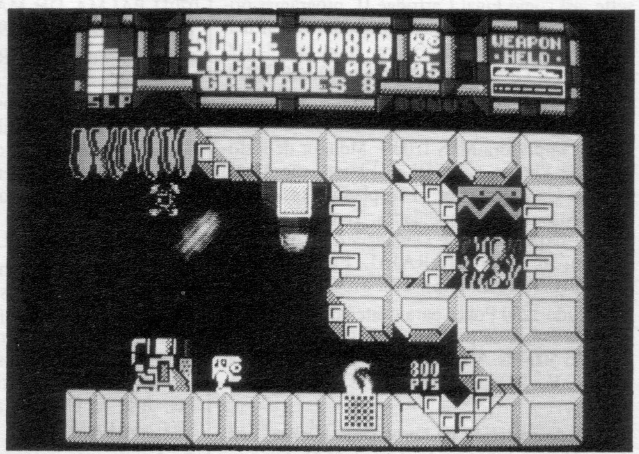
Los DROIDS es una graciosa adaptación de la conocida STARWARS, sólo que aquí, nuestros amigos R2-D2 y C-3PO se enfrentan solos con una terrible banda de ladrones, los Fromms. Estos, después de haberse escapado de una supercárcel estelar de alta seguridad, han ido a parar al planeta Aurel en el que están los pequeños robots. Los Fromms cogieron a R2-D2 y a C-3PO y los encerraron en sus mazmorras. Ahora tienen que escapar.

Los controles de este juego son un poco incómodos, ya que se encuentran en la parte inferior de la pantalla, flecha a la izquierda movimiento hacia la izquierda de los robots etc. Los gráficos y el sonido no son buenos pero el juego en general está bien.

Blackwiche es un conocido juego, que se desarrolla en un barco sobre el que tú diriges al personaje, que guiado por su curiosidad recorre el barco en busca de aventuras. El barco pertenecía al Duque de Blackwiche, que tras iniciar un largo viaje hacia una isla desconocida, desapareció con toda la tripulación y con el barco. El arriesgado aventurero es Sir Arthur Pendragón, que se tendrá que enfrentar con la maldad que había en la isla con que se encontró Blackwiche. Con la espada que encontrarás, Sir Arthur podrá luchar con sus enemigos y así lograr su objetivo.

Por último Bosconian. Una supernave dirigida contra los invasores de la tierra, tiene que recorrer el espacio eliminando a todos los enemigos. La máxima dificultad del juego, radica en la velocidad y en los muchos obstáculos con los que tu nave puede chocar. Distribuidas por el espacio, se encuentran letras con las que se consigue más energía etc. ■





MARKETCLUB

Servicio gratuito para nuestros lectores *PARTICULARES*. Los anuncios serán publicados durante 1 mes.
Los anuncios gratuitos de Market Club *SOLAMENTE* serán publicados si vienen con nombre, apellidos y dirección completa.

MERCADILLO

- Estoy interesado en intercambiar experiencias del sistema operativo GEOS, gráficos en general. También video y teleprints. Interesados podéis contactar con Agustin Oset. Bofarull, 42 4º2ª; 08027-BARCELONA. Telf.: 3524017.
- Vendo RATON C-64/128 con Software en Cassette por 7000 Pts. Juan C. Piña Perez. Casal Aboy, 13 2ºDcha. 36700- TUY(PONTEVEDRA). Tlf.: 601695.
- Vendo impresora Seikosha SP-1000VC con muy poco uso, como nueva. Totalmente compatible Commodore, se conecta al bus de serie. Múltiples tipos de letra: itálica, expandida, comprimida... Calidad NLQ. Impresión de gráficos. Fricción o tracción. Con manuales en inglés y castellano. Cables de conexión. Todo por 28.000 Pts. más gastos de envío. Jose E. Rodriguez Puertas. Avda. Barcelona, 7. 08700-IGUALADA (BARCELONA). Tlf.: 8033472. Llamar por las noches.
- Vendo C-128, Datasette 1530, FC-II, dos joysticks, juegos y revistas. Todo en perfecto estado por 43000 Pts. Adan Fernandez Oviaño. Suarez Inclan, 14 1ºIzda.; CUDILLERO (ASTURIAS). Tlf.: 590362.
- Compro unidad de disco 1541, preferiblemente en Tarragona. Jose R. Garcia Franco. Benidorm, 3-5A 4E. TERREVS. TARRAGONA. Tlf.: 313989.
- Vendo C-64, cassette, dos joysticks »Quitshot II«, curso completo de programación, libro CAD/CAM y varios juegos. Javier Belmonte Martin. Buenos Aires, 12-14 8-2. 08029-BARCELONA. Tlf.: 4107901.
- Vendo modulador de TV. para conectar el »Amiga 500«, a un TV. Como nuevo. Por 4.000 Pts. Oscar Tarruella Ayza. Pzo. d'en Llull N:12-3dcha. 08392-San Andres de Llavaneras. Barcelona. Tlf.: 7927759.
- Vendo C-64 con unidad de discos 1541 y datasette, así como Final Cartridge II, programas originales de Base de Datos Procesadores con manuales y revistas Commodore, todo por 54.000 Pts. Juan J. Gomez. Apostol Santiago, 49. 28017-MADRID. Tlf.: 7479600 Ext. 2611.
- Vendo ordenador C-64 + Unidad de disco 1541 + Impresora MPS 1200 + Monitor de fósforo verde Philips + FC-III + El cartucho Magic Desk + Un órgano que va acoplado al ordenador + Revistas, juegos y utilidades. Todo en perfecto estado. Expedito Jaime Dominguez. Avda. Constitución, 36. GRAN TARAJAL. FUERTEVENTURA (LAS PALMAS). Tlf.: 870038.
- Vendo 1250 Pokes para centenares de juegos, más 650 Pokes de programación (para C-64/128), por solo 500 Pts. Mandar a Fco. Exposito. Avda. Carlos V, 24 8B. 28936-MOSTOLES (MADRID).
- Vendo colección de libros Data Becker: Lenguaje Máquina Avanzado, Floppy 1541, Gráficos para C-64, 64 Interno, Pics í Pokes, Juegos y Trucos para el C-64, etc... por 12.000 pts. Juan Jose Gomez. Apostol Santiago, 49. 28017-Madrid. Tlf.:7479600 Ext. 2611.
- Vendo impresora RITEMAN C+ NLQ en perfecto estado, 35000 Pts. Regalo interface Buffer Stack y juegos para el Commodore 64 así como algunos libros de programación para este ordenador. Javier Guerrero Diaz. Emilio Santacana, 5. 11201-ALGECIRAS-CADIZ. Tlf.: 956-630643.
- Vendo libros para el C-64 por cambio de ordenador: 64 Interno, Lenguaje Máquina para Avanzados, etc... También algunos libros de informática y electrónica. David L. García Molina. Antonio López, 51 9D. 28019-MADRID. Tlf.: 4694626.
- Vendo C-128, 30000 Pts. Unidad de disco 1571, 30000Pts. Si compras todo te regalo impresora MPS-801, cassette, joystick, programas de utilidades y juegos. Jose Casais. Ponte de Saa, 74. 27800-VILLALBA-LUGO.
- Vendo C-64, 1571, datasette, monitor (HANTAREX), joystick, revistas, libros Data Becker y discos con programas variados (todos con instrucciones). Precio a convenir. Jim Juncosa Ollé. Marqués de Mulhacén, 2 6I. 08034-BARCELONA. Tlf.: 2039424.
- Vendo C-64, datasette, unidad de discos 1541, impresora Super Riteman F+, Final C. II, joysticks, Discos, Cassettes, funda para impresora, interface copiador. Todo en perfecto estado, por el precio de 80000 Pts. Luis Santapau Salvador. Cartagena, 293 Bis At.1.08025-BARCELONA. Tlf.: 2557291.
- Compact Disc marca Philips con pletina, cassette y radio, acepta compact singles, todo STEREO Y PORTATIL. Lo cambiaría por una unidad de disco para Commodore 64+ Programación. Jose Guinart. Patxillardegui, 10 6b.20014-LOIOLA-SAN SEBASTIAN (GUIPUZCOA). Tlf.: 371313.
- Vendo C-64, unidad de discos 1571, cassette 1531, joystick, cartucho Final C. III, revistas y juegos. Valorado en 140000 Pts. lo vendo por 97000 Pts. David Felipe Villa. Can Valero, 18. 07011-PALMA DE MALLORCA-BALEARES. Tlf.: 971-296418 y 724895.
- Vendo impresora ROBOTRON, para C-64, sin apenas uso, utiliza carretes de máquina de escribir (super económicos), mínimo nivel de ruido, rápida y cómoda de utilizar. Admite papel continuo y hojas sueltas. Todo por 45000 Pts. NEGOCIABLES. También vendo unidad de discos 1571 para C-64/128. Poco uso. Se acompaña de programas. Solo 30.000 Pts. Ramón Fregda Navarro, Pza. la Sol, 9. 25007-LLEIDA
- Vendo revistas Commodore World desde la 1 hasta la actual. 100 Pts. revista. También vendo lote de libros: Lenguaje Máquina para C-64, 64 Trucos y Consejos, Programación avanzada del C-64, Forth para micros y un libro sobre impresoras. Todo el lote de libros (Data Becker) por 6000 Pts. Jose T. Cuello García, Arriba, 9 y 11 5D. 37002-SALAMANCA. Tlf.: 218754.
- Vendo bien cuidados, con sus embalajes y libros, C-128 (28000 Pts.). Regalo: Final C. II y juegos. Unidad de disco 1571 (30000 Pts). Regalo: Loto Super-pro, Contabilidad personal, etc... Riteman C+ (35000 Pts.) con sus libros. Monitor Commodore 1900 (14000 Pts.) con 4 meses, regalo cable 40/80 Columnas. Federico González Zárate. Nueva Torrevieja, 218. TORREVIEJA-ALICANTE. Tlf.: 96-5716418, llamar de 9,30 a 10,30 h. NOCHE.
- Vendo ordenador C-128, datassette, joystick, cintas, libros Data-Becker, revistas y disco CP/M, todo en perfecto estado. Precio a convenir. Miguel Angel Rojo. Torres Quevedo, 9 11C. 39011-SANTANDER. Tlf.: 942-332424. Llamar a partir 21h.
- Vendo impresora RITEMAN C+, más procesador de textos y base de datos. Todo 25000 Pts. Xavier Puigdomenech García. Dos de Maig, 204 5-2. 08013-BARCELONA.

CLUBS

- Amiga 500-1000-2000. El club Lethal boys ha abierto la recepción de nuevos socios. Inscripción gratuita y sin cuotas posteriores, programa regalo de bienvenida. Recibe toda la información que desees en: LETHAL BOYS CORP. MANUEL DE FALLA, 11 4C. 24400-PONFERRADA-LEON.

Hemos formado en Galicia un club de usuarios de C-64/128 para intercambio de programas en Galicia y fuera de ella. Club Raspacho Soft. Pablo Iglesias, 4 4º. 36600-VILLAGARCIA DE AROSA. PONTEVEDRA. Tlf.: 986-503120 preguntar por Juan.

- CLUB USUARIS COMMODORE-TARRAGONA, centro homologado por l Generalitat de Catalunya. Pídenos información de nuestras secciones de: Biblioteca, Programoteca, Cursos, Asesoramiento, Comunicaciones, etc... Seis años al servicio de nuestros socios. Apartado de Correos, 176. 43080-TARRAGONA. Local Social, Fortuny, 4 2-2. 43001-TARRAGONA.

CONTACTOS

- Desearía contactar con usuarios o bien clubs de C-64 en la zona de Gerona. Josep M. Rodríguez. Conill, 20 at.2. 17820- BANYOLES (GERONA) Tlf.: 574858.
- Intercambio todo tipo de información sobre AMIGA 500. Jose M. Campos Martinez. Antonio Palacios, 58 2A. 36400-PORRIÑO (PONTEVEDRA) Tlf.: 986-335809.
- Deseo contactar para intercambio de ideas, proyectos y programas, escribir a: Julio Espinosa Mercado. Francisco Hernandez Guerra, 8 7C. 35014-LAS PALMAS.
- Deseo contactar con amigos del C-64. Juan Martin Gril. Avda. Pueyrredon, 923 12A. 1032-BUENOS AIRES (CAPITAL FEDERAL) ARGENTINA. Tlf.: 962-2379.

DISCOS AMIGA WORLD 5



Ya está a la venta el quinto disco Amiga World, con los programas que periódicamente publicamos en la sección para Amiga dentro de Commodore World. Este disco corresponde a los números 58, 59 y 60 de la revista. Como en otras ocasiones, incluye todos los programas que han aparecido listados, así como otros de regalo. Como programa especial se incluye la ampliación **Gadgets Basic,** un magnífico programa que puede ser de gran utilidad para todos los usuarios que emplean el Amiga Basic como lenguaje de programación y además quieren hacer uso de las «intuitivas» posibilidades del Amiga. El disco incluye también los últimos listados del **Curso de C,** los del **Curso de Ensamblador** y otros programas de demostración y utilizades.

POR SOLO 1.700 PESETAS, EL DISCO CON TODOS LOS PROGRAMAS DE LAS REVISTAS 58, 59 Y 60.

BOLETIN DE PEDIDO - DISCOS AMIGAWORLD V

Nombre ....

Dirección ....

Población .... C.P .... Provincia ....

Teléfono .... Modelo de Amiga ....

☐ Deseo recibir el disco AMIGAWorld 5 (1.700 ptas.)

☐ Deseo suscribirme por un año (4 discos y 11 revistas) al precio especial de 9.000 ptas., a partir de la revista número ....

☐ Incluyo cheque por .... ptas.

☐ Incluyo giro número .... por .... ptas.

Enviar a: COMMODORE WORLD, Rafael Calvo, 18, 4.º B. 28010 Madrid.

Forma de pago: Sólo cheque giro. No se sirven pedidos contrarreembolso. Gastos de envío incluidos.

La suscripción se puede pagar por tarjeta VISA o MASTERCARD, bien por carta o por teléfono.



INICIACION AL LENGUAJE
ENSAMBLADO

AMIGADOS Y EXEC LIBRARY — 2ª Part

Este capítulo segundo va a ocuparse del AmigaDOS, es decir, la librería encargada de manejar ficheros, ya sea leerlos dato a dato, escribir, borrar, renombrar, abrir, cerrar... Esto es válido para cualquier periférico del AMIGA, por ejemplo la pantalla o la impresora.

Por Fernando García

La única ayuda que se necesita es saber que algunas veces hay que acceder a algunas funciones de otra librería, que veremos ahora. La librería EXEC, también conocida como la librería del sistema, es la primera librería a la que podemos acceder cuando se realiza un programa. Si recuerdas el anterior capítulo, para abrir una librería había que acceder a la función OpenLibrary de la misma librería EXEC. ¿Cómo vamos a abrir una librería si no podemos acceder a la función? La solución para el ordenador sería que al encenderse situara la base de esta librería en una posición fija. Así sucede, exactamente en la posición $4. Resumiendo, la manera más fácil de abrir una librería es como se ve en el listado 1.

Ahora que ya sabemos dónde está la librería EXEC vamos a explicarla un poco además de algunas de sus funciones que necesitamos para manejar el AmigaDOS (dos.library).

La librería Exec puede considerarse como la librería más utilizada, por eso es tan fácil acceder a ella,, y contiene funciones muy importantes, bien sean de manejo de memoria, de interrupciones, de señales, de puertos, de periféricos... Ocuparía más de un capítulo explicar todo esto. Por ahora veremos solamente la función *AllocMem,* encargada de ocupar o reservar un espacio de memoria. Sus parámetros son éstos:

Bloque de memoria = AllocMem (longitud, requerimientos)
D0 D0 D1

El Offset exacto de esta función es -$C6 en hexadecimal. Antes de acceder a ella pondremos en sus respectivos registros la longitud de memoria que queremos reservar para uso propio y los requerimientos de este bloque de memoria. Una vez hecha la llamada a la función, ésta devolverá en el registro D0 la posición en la que se ha reservado la memoria. Los requerimientos indican qué tipo de memoria va a ser. Los tipos disponibles son:

MEMORIA PUBLICA 1
Cualquier tipo de memoria, con preferencia a FAST RAM
CHIP MEMORY 2
Los primeros 512k de RAM
FAST MEMORY 4
RAM de expansión
MEMORIA BORRADA
RAM que ha sido inicializada con ceros-
MEMORIA LARGA 20000
El bloque más larga contiguo de RAM

Hay dos opciones a elegir, por una parte el tipo de localización de RAM



(los tres primeros valores), y la manera de inicializar la memoria (los dos últimos). A continuación se suman tdos estos valores, y ese es el resultado del parámetro que hay que insertar en el registro D1. Hay que tener en cuenta algunas cosas, por ejemplo, para reservar (Allocate) un bloque de memoria que nos sirva como pantalla gráfica hay que elegir CHIP MEMORY (2) y MEMORIA BORRADA (10000). Al sumar estos dos valores, el parámetro final es 2+10000=10002. Sin embargo si vas a utilizar el bloque de memoria para guardar datos de cualquier tipo escoge FAST MEMORY y MEMORIA BORRADA (4+10000 =10004), para ahorrar CHIP MEMORY, que será más valiosa después. No se pueden elegir dos tipos de memoria a la vez.

Una vez ejecutada la función, devolverá en D0 el lugar exacto en donde se encuentre el bloque de memoria, que hay que salvar a continuación en un puntero. Un ejemplo para reservar 5000 bytes de FAST MEMORY puede verse en el listado 2.

Ahora podrás utilizar tu bloque de memoria para uso propio, teniendo el AMIGA total conocimiento de que la tienes reservada, haciendo referencia al puntero «MEMORIA».

Como dijimos en el capítulo anterior que todo lo que abramos antes de salirnos del programa hay que cerrarlo, veamos cómo se libera este bloque de memoria para que el AMIGA pueda utilizarlo posteriormente. Esto se hace con la función contraria, es decir, con FreeMem. Su Offset es el -$D2 y los parámetros son los siguientes:

FreeMem (puntero del bloque de memoria, longitud)
A1 D0

Para desocupar el bloque de memoria del ejemplo anterior, ver listado 3.

Hay varias funciones más de la librería Exec que te permiten reservar memoria, pero no es preciso explicarlas ahora pues con esto es suficiente para este capítulo.

AmigaDOS y los ficheros

Empecemos ahora con el AmigaDOS, que realiza una ardua tarea en el AMIGA, permitiendo escribir datos de cualquier tipo a todos los periféricos instalados internamente, así como de recibir mensajes de otros periféricos (teclado, serial, etcétera). El completo conjunto de funciones de que constan están explicadas en el libro AmigaDOS Manual. Veamos ahora las funciones más importantes disponibles en esta librería:

INPUT

Estructura:
Fichero=Input( )
Registros:
D0
Resultado:
El fichero actual (estándar) de entrada
Descripción:
Permite obtener el fichero actual por donde van a salir nuestros mensajes, que la mayoría de las veces es la ventana del CLI.

OUTPUT

Estructura:
Fichero=Output( )
Registros:
D0
Resultado:
El fichero actual de salida
Descripción:
Obtiene el fichero de entrada de datos, que es también el CLI.

OPEN

Estructura:
Fichero=Open(nombre, modo de acceso)
Registros:
D0 D1 D2

Parámetros:
Nombre = puntero del nombre terminado con byte 0
Modo acceso = tipo de acceso al abrir el fichero
Resultado:
Estructura del fichero
Descripción:
Los modos de acceso a la hora de abrir un fichero pueden ser varios. Si es MODO ANTIGUO FICHERO (=1005) abre un antiguo fichero para posteriormente leer o escribir datos. También puede ser MODO NUEVO FICHERO (=1006) creando uno nuevo para su posterior utilización. La escritura de estos ficheros puede hacerse a diversos periféricos como RAW:, NIL:, o CON:, etcétera.

CLOSE

Estructura:
Close (fichero)
Registros:
D1
Descripción:
Cierra un fichero después de haber sido utilizado. El único parámetro utilizado es la estructura del fichero que se utiliza a la hora de abrir el mismo fichero.

READ

Estructura:
Longitud actual = Read (fichero, buffer, longitud)
Registros:
D0 D1 D2 D3
Parámetros:
Fichero = estructura del fichero
Buffer = un puntero al buffer Longitud = longitud de bytes a leer
Resultado:
Se obtiene la longitud del número de bytes leídos
Descripción:
Permite leer datos de un fichero hacia el buffer con una longitud especificada. El valor devuelto es el número de bytes leídos, permitiéndote saber si ha llenado el buffer (fin de fichero o entrada). Si el valor devuelto es -l ha ocurrido un error. Para más información del tipo de error ver la función IoErr( ).

WRITE

Estructura:
Longitud retorno = Write (fichero, buffer, long.)

```
START     MOVEA.L  4,A6              Apunta a Exec.library
          LEA      DOSNAME,A1        Apunta al nombre
          JSR      -$198(A6)         Abre librería
          MOVE.L   D0,DOSBASE        Salva base librería
          .
DOSNAME   DC.B     'dos.library',0   Nombre de la librería
          CNOP     0,2               Alinea a WORD lo siguiente
DOSBASE   DS.L     1                 Almacenamiento base librería
```

Listado 1

```
          MOVEA.L  4,A6              Apunta a Exec.library
          MOVE.L   #5000,D0          5000 bytes a reservar
          MOVE.L   #$10004,D1        Memoria borrada (10000) y fast RAM (4)
          JSR      -$C6(A6)          Ejecuta función AllocMem
          MOVE.L   D0,MEMORIA        Salva el puntero de la memoria
          BEQ      MEMERROR          Saltar si hay error
          .                          Resto de programa
          .
MEMORIA   DS.L     1                 Posición de la memoria
```

Listado 2

```
          MOVEA.L  4,A6              Apunta a Exec.library
          MOVE.L   #5000,D0          5000 bytes para desocupar en D0
          MOVE.L   MEMORIA,A1        Salva puntero del bloque en A1
          JSR      -$D2(A6)          Función FreeMem
```

Listado 3

```
          MOVE.L   4(A5),D1          apunta al nombre del nuevo fichero
por       MOVE.L   #NEW_F,D1         apunta al nombre del nuevo fichero

e insertar en el bloque de mensajes la linea

NEW_F     DC.B     "CON:0/20/640/150/FICHERO",0
```

Listado 4

```
          MOVE.L   FILEHANDLE,D1     Obtiene estructura del fichero
          MOVE.L   #0,D2             Posición 0
          MOVE.L   #-1,D3            Offset COMIENZO (=-1)
          JSR      -$42(A6)          Función seek

      *   Para situarnos al final d

          MOVE.L   FILEHANDLE,D1     Obtiene estructura del fichero
          MOVE.L   #0,D2             Posición 0
          MOVE.L   #1,D3             Offset FINAL (=1)
          JSR      -$42(A6)          Función seek

      *   para situarnos 100 bytes del comienzo
          del fichero:

          MOVE.L   FILEHANDLE,D1     Obtiene estructura del fichero
          MOVE.L   #100,D2           Posición 100
          MOVE.L   #-1,D3            Offset COMIENZO (=-1)
          JSR      -$42(A6)          Función seek
```

Listado 5

Registros:
D0 D1 D2 D3

Parámetros:
Los mismos que en la función READ

Resultado:
Longitud de la información actualmente escrita

Descripción:
Escribe datos a un fichero, como buffer el puntero de datos a transmitir y la longitud del número de bytes. Si la función devuelve un -1, quiere decir que ha ocurrido algún error. El error exacto se puede averiguar con la función IoErr( ).

LOCK

Estructura:
Lock = Lock (nombre, modo de acceso)

Registros:
D0 D1 D2

Parámetros:
Nombre = nombre del fichero o directorio
Modo de acceso = igual que READ y WRITE
que READ y WRITE

Resultado:
Lock, bloqueo o llave del fichero

Descripción:
La función permite obtener un bloqueo (lock) del fichero o directorio para su posterior utilización en funciones como EXAMINE. Tiene dos modos de acceso, Read (-2) y Write (-1) dependiendo de la utilización del fichero. Si la función se lleva a cabo devolverá la estructura del bloque en D0; si no, devolverá un 0 o un nulo.

UNLOCK

Estructura:
Unlock (lock)

Registros:
D1

Parámetros:
Lock = estructura del bloqueo

Descripción:
Se utiliza simplemente para desbloquear el fichero y antes de cerrar el fichero

EXAMINE

Estructura:
Suceso = Examine (lock, Bloque Info del fichero)

```
              XREF    paramadd
              XREF    _DOSBase
              XREF    _stdin
              XDEF    START
;OBTIENE LOCK DEL FICHERO
START         MOVE.L  _DOSBase,A6          Apunta a la base librería DOS
              LEA     paramadd,A5          Apunta parámetros
              MOVE.L  (A5),D1              Apunta al nombre
              MOVE.L  #-2,D2               Modo de acceso READ
              JSR     -$54(A6)             Bloquea fichero (función LOCK)
              MOVE.L  D0,RLOCK             Salva el bloqueo
              BEQ     ERROR_A              Salir si hay error
; ABRE EL FICHERO
              MOVE.L  (A5),D1              Apunta al fichero
              MOVE.L  #1005,D2             Modo de acceso ANTIGUO FICHERO
              JSR     -$1E(A6)             Abre fichero (función OPEN)
              MOVE.L  D0,RHANDLE           Salva estructura fichero (filehandle)
              BEQ     ERROR_B              Salir si se produce error
; OBTIENE BLOQUE INFO
              MOVE.L  RLOCK,D1             Salva bloqueo (LOCK) en D1
              MOVE.L  #INFOBLKBUFF,D2      Apunta al buffer
              JSR     -$66(A6)             Examina (función EXAMINE)
              TST.L   D0                   Comprueba si hay error
              BEQ     ERROR_C              Si, salta a salida
              LEA     INFOBLKBUFF,A0       Apunta al buffer
              MOVE.L  124(A0),D7           Obtiene longitud fichero
; RESERVA MEMORIA PARA ALMACENAR EL FICHERO
              MOVE.L  D7,D0                Salva longitud fichero en D0
              MOVE.L  #$10002,D1           Tipo de memoria (MEMORIA CHIP)
              MOVE.L  4,A6                 Apunta a base librería Exec
              JSR     -$C6(A6)             Reserva memoria (función ALLOCMEM)
              MOVE.L  _DOSBase,A6          Apunta a base librería DOS
              MOVE.L  D0,MEMBLK            Salva el puntero del bloque reservado
              BEQ     MEMERROR             Saltar si se ha producido error
; LEE TODO EL FICHERO
              MOVE.L  RHANDLE,D1           Obtiene estructura fichero (filehandle)
              MOVE.L  MEMBLK,D2            Apunta al bloque de memoria del fichero
              MOVE.L  D7,D3                Obtiene longitud fichero
              JSR     -$2A(A6)             Lee todo el fichero (función READ)
              CMP.L   D0,D7                Comprueba todos los bytes leidos
              BNE     ERROR_D              Si no, hay error y sale del programa
              MOVE.L  D0,FLONG             Si, pues salva longitud
; ABRE EL FICHERO
              MOVE.L  4(A5),D1             Apunta al nombre del nuevo fichero
              MOVE.L  #1006,D2             Modo de acceso NUEVO FICHERO
              JSR     -$1E(A6)             Abrir el fichero (OPEN)
              MOVE.L  D0,WHANDLE           Salvar estructura fichero
              BEQ     ERROR_E              Saltar si hay error
; ESCRIBE SOBRE EL NUEVO FICHERO
              MOVE.L  WHANDLE,D1           Obtiene estructura fichero
              MOVE.L  MEMBLK,D2            Apunta al bloque de memoria (fichero)
              MOVE.L  FLONG,D3             Obtiene longitud fichero
              JSR     -$30(A6)             Escribe todos los bytes (WRITE)
              CMP.L   FLONG,D0             Comprobar si se ha escrito completo
              BNE     ERROR_F              No, saltar a salida programa
              MOVE.L  #MSJ_FINAL,D2        Apunta comienzo mensaje
              MOVE.L  #MSJ_FINALE,D3       Apunta al final mensaje
              BSR     PRTMENS              Enviar mensaje a CLI
              BRA     EXIT1                Salta a rutina de salida
; RUTINAS DE ERROR
ERROR_A       MOVE.L  #ERRORMSJ1,D2        Apunta comienzo mensaje
              MOVE.L  #ERRORMSJ1E,D3       Apunta al final mensaje
              BSR     PRTMENS              Enviar mensaje a CLI
```

Listado 6

Registros:
D0 D1 D2
Parámetros:
Lock = lock o bloqueo del fichero o directorio Bloque Info = estructura de 264 bytes alineados a palabra larga
Resultado:
Devolverá un 0 en el registro D0 si falla la función
Descripción:
Permite obtener la estructura del bloque de información acerca del fichero o directorio. Este bloque ha de ser alineado a palabra larga con la estructura CNOP y con el valor 0,4. Una vez realizada la función se puede obtener información tan valiosa como longitud del fichero, número de bloques, protecciones, etcétera. En el programa que acompaña a este capítulo pondremos algún valor en concreto.

IOERR

Estructura:
Error = IoErr( )
Registros:
D0

Resultado:
Error = tipo de error ocurrido
Descripción:
Si alguna vez obtienes algún error al intentar acceder a un fichero con esta función podrás saber el tipo de error ocurrido. El set de errores está en la tabla adjunta a este capítulo

SEEK

Estructura:
Posición antigua = Seek (fichero, posición, modo)
Registros:
D0 D1 D2 D3
Parámetros:
Fichero = estructura del fichero (filehandle)
Posición = nueva posición del fichero
Modo = modo de posición del fichero
Resultado:
Posición antigua = posición antes de realizar la función
Descripción:
La función posiciona el »cursor« del fichero para su posterior lectura o escritura del mismo. Permite por ejemplo juntar datos con el final del fichero. Hay tres modos de posicionar ese cursor: COMIENZO (-1),

```
            BRA     EXIT4               Salida de programa
ERROR_B     MOVE.L  #ERRORMSJ2,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJ2E,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT3               Salida de programa
ERROR_C     MOVE.L  #ERRORMSJ3,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJ3E,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT2               Salida de programa
MEMERROR    MOVE.L  #ERRORMSJM,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJME,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT2               Salida de programa
ERROR_D     MOVE.L  #ERRORMSJ4,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJ4E,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT5               Salida de programa
ERROR_E     MOVE.L  #ERRORMSJ5,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJ5E,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT5               Salida de programa
ERROR_F     MOVE.L  #ERRORMSJ6,D2       Apunta comienzo mensaje
            MOVE.L  #ERRORMSJ6E,D3      Apunta al final mensaje
            BSR     PRTMENS             Enviar mensaje a CLI
            BRA     EXIT1               Salida de programa
PRTMENS     SUB.L   D2,D3               Obtiene longitud fichero con resta
            MOVE.L  _stdin,D1           Apunta a la salida standard (CLI)
            JSR     -$30(A6)            Escribe en cli mensaje (func. WRITE)
            RTS                         Vuelve al CLI
; RUTINAS DE SALIDA
EXIT1       MOVE.L  WHANDLE,D1          Obtiene estructura fichero
            JSR     -$24(A6)            Cierra fichero (función CLOSE)
EXIT5       MOVE.L  MEMBLK,A1           Apunta al bloque de memoria
            MOVE.L  FLONG,D0            Obtiene longitud bloque memoria
            MOVE.L  4,A6                Apunta base librería Exec
            JSR     -$D2(A6)            Desocupa memoria (función FREEMEM)
            MOVE.L  _DOSBase,A6         Apunta a base librería DOS
EXIT2       MOVE.L  RHANDLE,D1          Obtiene estructura fichero
            JSR     -$24(A6)            Cierra fichero (func. CLOSE)
EXIT3       MOVE.L  RLOCK,D1            Obtiene bloqueo (LOCK)
            JSR     -$5A(A6)            Desbloqueo (función UNLOCK)
EXIT4       RTS                         Vuelve al CLI
; PUNTEROS
            CNOP    0,2                 Alinea punteros
WHANDLE     DS.L    1
RHANDLE     DS.L    1
RLOCK       DS.L    1
FLONG       DS.L    1
MEMBLK      DS.L    1
; MENSAJES
ERRORMSJ1   DC.B    'No puedo encontrar fichero antiguo',$A
ERRORMSJ1E  DC.B    0
ERRORMSJ2   DC.B    'No puedo abrir antiguo fichero',$A
ERRORMSJ2E  DC.B    0
ERRORMSJ3   DC.B    'No puedo examinar antiguo fichero',$A
ERRORMSJ3E  DC.B    0
ERRORMSJM   DC.B    'No hay memoria disponible',$A
ERRORMSJME  DC.B    0
ERRORMSJ4   DC.B    'Error de lectura',$A
ERRORMSJ4E  DC.B    0
ERRORMSJ5   DC.B    'No puedo abrir nuevo fichero',$A
ERRORMSJ5E  DC.B    0
ERRORMSJ6   DC.B    'Error de escritura',$A
```

ACTUAL (0), y FINAL (1). Se puede adelantar o atrasar el número de bytes indicados en el registro D2 (incluso el signo) a partir del comienzo, final o posición actual, para acceder o escribir sobre el fichero.

La librería tiene más funciones, pero por ahora con éstas es suficiente para la creación y acceso a un fichero. De todas formas adjuntamos una tabla con todas las funciones, además de los offsets que no hemos añadido a las que hemos explicado. Vamos a resumir fácilmente cómo se lee un fichero (no un programa):

1. Se obtiene un **bloqueo del fichero** con la función LOCK. Para ello necesitaremos saber el nombre del fichero y el modo de acceso (READ = -2).
2. Se abre el fichero. Función OPEN.
3. Se **examina** el fichero para conocer la longitud del mismo. Función EXAMINE.
4. Se reserva tanta **memoria** como indique la longitud del fichero con la función ALLOCMEM.
5. Se lee todo el fichero. Función READ.

Detrás de cada función se pueden incluir rutinas para averiguar si ha ocurrido algún error en el proceso (función Ioerr). Ahora ya está dispuesto el fichero para ser examinado y una vez terminado este proceso hay que cerrar todo lo que se haya abierto. Cualquier proceso mal llevado a cabo producirá un Guru. Para cerrarlo todo se sigue el proceso contrario.

1. Se *libera* la memoria del fichero con la función FREEMEM.
2. Se *cierra* el fichero. Función CLOSE.
3. Se *desbloquea* el fichero. Función UNLOCK.

Este es el proceso completo para acceder a un fichero. Con pocos cambios más se puede acceder a la información de un directorio, añadir datos, etcétera.

En la tabla 1 de funciones se combinan varias columnas. Las tres primeras columnas corresponden a los offsets en sus diferentes numeraciones de decimal hexadecimal absoluto (sin signo) y al valor en hexadecimal negativo (comúnmente utilizado). La siguiente columna corresponde a la función con los parámetros entre paréntesis y la cuarta columna son los registros correspondientes a los parámetros anteriores en su orden correcto. He de decir que esta tabla corresponde a la del apéndice D del libro ROM Kernel Reference Manual: Exec, cuyos datos son erróneos; por tanto la tabla sirve como co-

```
ERRORMSJ6E      DC.B    0
MSJ_FINAL       DC.B    'Copia de fichero completo',$A
MSJ_FINALE      DC.B    0
; BUFFER BLOQUE INFO
                CNOP    0,4          Alinea bloque info en palabra larga
INFOBLKBUFF     DS.B    264
                END
```

Listado 6

Dec.	Hexadecimal Absol.	Neg.	Función (parámetros)
30	FFE2	-1E	OPEN (NOMBRE, MODO ACCESO) (D1,D2)
36	FFDC	-24	CLOSE (FICHERO) (D1)
42	FFD6	-2A	READ (FICHERO,BUFFER,LONGITUD) (D1,D2,D3)
48	FFD0	-30	WRITE (FICHERO,BUFFER,LONGITUD) (D1,D2,D3)
54	FFCA	-36	INPUT ()
60	FFC4	-3C	OUTPUT ()
66	FFBE	-42	SEEK (FICHERO, POSICION, MODO) (D1,D2,D3)
72	FFB8	-48	DELETEFILE (NOMBRE) (D1)
78	FFB2	-4E	RENAME (ANTIGUO NOMBRE, NUEVO NOMBRE) (D1,D2)
84	FFAC	-54	LOCK (NOMBRE, TIPO) (D1,D2)
90	FFA6	-5A	UNLOCK (LOCK) (D1)
96	FFA0	-60	DUPLOCK (LOCK) (D1)
102	FF9A	-66	EXAMINE (LOCK, BLOQUE INFO) (D1,D2)
108	FF94	-6C	EXNEXT (LOCK, BLOQUE INFO) (D1,D2)
114	FF8E	-72	INFO (LOCK, BLOQUE PARAMETRO) (D1,D2)
120	FF88	-78	CREATEDIR (NOMBRE) (D1)
126	FF82	-7E	CURRENTDIR (LOCK) (D1)
132	FF7C	-84	IOERR ()
138	FF76	-8A	CREATEPROC (NOMBRE, PRIOR, LISTA SEGM, LONG STACK) (D1,D2,D3,D4)
144	FF70	-90	EXIT (CODIGO RETORNO) (D1)
150	FF6A	-96	LOADSEG (SEGMENTO) (D1)
156	FF64	-9C	UNLOADSEG (SEGMENTO) (D1)
162	FF5E	-A2	GETPACKET (WAIT) (D1)
168	FF58	-A8	QUEUEPACKET (PACKET) (D1)
174	FF52	-AE	DEVICEPROC (NAME) (D1)
180	FF4C	-B4	SETCOMMENT (NOMBRE, COMENTARIO) (D1,D2)
186	FF46	-BA	SETPROTECTION (NOMBRE, MASCARA) (D1,D2)
192	FF40	-C0	DATESTAMP (FECHA) (D1)
198	FF3A	-C6	DELAY (TIEMPO) (D1)
204	FF34	-CC	WAITFORCHAR (FICHERO, TIEMPO) (D1,D2)
210	FF2E	-D2	PARENTDIR (LOCK) (D1)
216	FF28	-D8	ISINTERACTIVE (FICHERO) (D1)
222	FF22	-DE	EXECUTE (COMANDO,INPUT,OUTPUT) (D1,D2,D3)

Tabla 1. Funciones

VALOR	ERROR
103	No más almacenamiento libre
202	Objeto en uso
203	Objeto ya existe
205	Objeto no encontrado
209	Acción desconocida
210	Componente del nombre inválido
211	Bloqueo (lock) inválido
212	Tipo de objeto incorrecto
213	Disco no válido
214	Disco protegido
215	Renombrar periférico
216	Directorio no vacío
218	Periférico no montado
219	Error de función SEEK
220	Comentario demasiado largo
221	Disco lleno
222	Protección de borrado de ficheros
223	Protección de escritura de ficheros
224	Protección de lectura de ficheros
225	No es un disco DOS
226	No hay disco
232	No más entradas de directorio

Tabla 2. Errores de la función Ioerr()

```
Listado 7
Programa: Genera.PRG

REM Programa generador "PRG"              . 13
REM (c)1989 by Fernando Garcia            . 27
REM (c)1989 by Commodore World            .261

CLS                                       .313
OPEN "PRG" FOR OUTPUT AS 1                .780
FOR a=1 TO 1868                           .177
READ b$                                   . 57
ca= ASC (LEFT$(b$,1))                     .720
ca=ca-48                                  . 75
IF ca>9 THEN ca=ca-7                      .786
REM PRINT  ca                             .977
ca=ca*16                                  .883
cb= ASC (RIGHT$(b$,1))                    .567
cb=cb-48                                  .845
IF cb>9 THEN cb=cb-7                      .264
cc=ca+cb                                  .722
PRINT #1,CHR$(cc);                        .666
NEXT a                                    .241
CLOSE 1                                   . 56
END                                       .992

DATOS:                                    .176
DATA 00,00,03,F3,00,00,00,00,00,00  .897
DATA 00,02,00,00,00,00,00,00,00,01  .329
DATA 00,00,00,8C,00,00,00,ED,00,00  .463
DATA 03,E9,00,00,00,8C,23,CF,00,00  .234
DATA 01,78,48,E7,80,80,2C,78,00,04  .498
DATA 23,CE,00,00,01,6C,93,C9,4E,AE  .632
DATA FE,DA,28,40,23,C0,00,00,01,74  .234
DATA 43,F9,00,00,02,24,4E,AE,FE,68  .402
DATA 23,C0,00,00,01,70,4A,AC,00,AC  .679
DATA 67,00,00,8C,42,82,45,F9,00,00  .986
DATA 01,D4,47,F9,00,00,01,AC,49,F9  .528
DATA 00,00,01,84,4C,DF,01,01,42,83  .918
DATA 12,18,04,80,00,00,00,01,6F,00  .352
DATA 00,36,0C,01,00,20,6F,EE,06,82  .818
DATA 00,00,00,01,06,83,00,00,00,01  .606
DATA 26,CA,14,C1,12,18,04,80,00,00  .167
DATA 00,01,0C,01,00,20,6F,00,00,0A  .841
DATA 06,83,00,00,00,01,60,E6,42,1A  .479
DATA 28,C3,60,BE,23,C2,00,00,01,80  .177
DATA 2C,79,00,00,01,70,4E,AE,FF,CA  . 85
DATA 23,C0,00,00,01,64,4E,AE,FF,C4  .224
DATA 23,C0,00,00,01,68,2C,79,00,00  .642
DATA 01,6C,4E,B9,00,00,00,00,60,00  .659
DATA 00,94,41,EC,00,5C,4E,AE,FE,80  .609
DATA 41,EC,00,5C,4E,AE,FE,8C,23,C0  .198
DATA 00,00,01,7C,42,A7,2F,00,24,40  . 27
DATA 20,2A,00,24,67,00,00,10,2C,79  .491
DATA 00,00,01,70,20,40,22,10,4E,AE  .204
DATA FF,82,22,2A,00,20,67,00,00,26  .930
DATA 24,3C,00,00,03,ED,4E,AE,FF,E2  .696
DATA 23,C0,00,00,01,64,23,C0,00,00  .469
DATA 01,68,67,00,00,0C,E5,88,20,40  .814
DATA 29,68,00,08,00,A4,2C,79,00,00  .330
DATA 01,6C,4E,B9,00,00,00,00,2C,79  .435
DATA 00,00,01,6C,22,79,00,00,01,70  .440
DATA 4E,AE,FE,62,4A,B9,00,00,01,7C  .137
DATA 67,00,00,10,4E,AE,FF,7C,22,79  .532
DATA 00,00,01,7C,4E,AE,FE,86,20,3C  .875
DATA 00,00,00,00,2E,79,00,00,01,78  .645
DATA 4E,75,00,00,00,00,00,00,00,00  .768
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,64,6F,73,2E,6C,69  .491
DATA 62,72,61,72,79,00,00,00,03,EC  . 16
DATA 00,00,00,17,00,00,00,00,00,00  .685
DATA 01,5E,00,00,01,4E,00,00,01,40  .900
DATA 00,00,01,36,00,00,01,30,00,00  .901
DATA 01,24,00,00,01,10,00,00,01,0A  .920
DATA 00,00,00,EA,00,00,00,D6,00,00  .372
DATA 00,B6,00,00,00,B0,00,00,00,A6  .784
DATA 00,00,00,9C,00,00,00,96,00,00  .510
DATA 00,4A,00,00,00,44,00,00,00,3E  .322
DATA 00,00,00,2E,00,00,00,24,00,00  .445
DATA 00,1E,00,00,00,10,00,00,00,02  .711
DATA 00,00,00,02,00,00,00,01,00,00  .968
DATA 00,BC,00,00,01,2A,00,00,00,00  .728
DATA 00,00,03,F2,00,00,03,E9,00,00  .221
DATA 00,ED,2C,79,00,00,01,70,4B,F9  .363
DATA 00,00,01,AC,22,15,24,3C,FF,FF  .729
DATA FF,FE,4E,AE,FF,AC,23,C0,00,00  .163
DATA 01,BE,67,00,00,C2,22,15,24,3C  .885
DATA 00,00,03,ED,4E,AE,FF,E2,23,C0  .206
DATA 00,00,01,BA,67,00,00,C0,22,39  .518
DATA 00,00,01,BE,24,3C,00,00,02,AC  .543
DATA 4E,AE,FF,9A,4A,80,67,00,00,BE  .417
DATA 41,F9,00,00,02,AC,2E,28,00,7C  .879
```

Listado 7

rrección. En definitiva, una llamada a la función WRITE se puede expresar de tres formas distintas:

JSR -30(A6)
JSR $FFE2(A6)
JSR -$1E(A6)

El programa

Veamos ahora el programa que ilustra todas estas funciones. Para que lo comprendas bien observa atentamente cómo se van insertando los parámetros correspondientes a cada función, cómo se hacen las llamadas y las verificaciones por si ha ocurrido algún error.

El programa del listado 6 es una simple utilidad de copia de un fichero. Una vez ensamblado basta poner desde CLI la línea PRG «fichero1» «fichero2» en el que fichero1 es el fichero inicial a copiar y fichero2 es el fichero duplicado y copiado. También se adjunta el programa en BASIC (listado 7) para los que no dispongan de ensamblador y quieran seguir los capítulos.

Una vez tecleado en el editor de textos, salvado a disco y ensamblado, prueba a copiar un fichero no muy largo, y si éste funciona, provoca intencionadamente algunos de los errores que se propone dentro del programa, poniendo un nombre incorrecto, llenando el disco, etcétera. Expliquemos el proceso de copia con tranquilidad.

Primero observa las primeras directivas, búscalas y compáralas con las del programa Astartup del primer capítulo. Especial mención merece «paramadd» que es la dirección de los parámetros que hemos entrado desde CLI, que en este caso son «fichero1» y «fichero2». Voy a explicar cómo se obtienen.

Cuando se carga un programa antes de ejecutarse el ordenador inserta en el registro A1 la dirección exacta del texto adicional a ese programa, es decir, los parámetros. Si ponemos «dir DF1:» el parámetro es «DF1:», entonces nuestro programa Astartup juntado con este último permite averiguar qué fichero es el que hay que copiar y adónde hay que copiarlo, mediante la dirección «paramadd». Esta dirección es en realidad un buffer que contiene las distintas direcciones de los distintos parámetros que acompañan a esta pequeña utilidad.

Ahora veamos cómo se conjuga con el programa. Primero obtenemos el bloqueo o «lock» del fichero a copiar. Para ello se necesitan dos parámetros o registros, el primero es la dirección



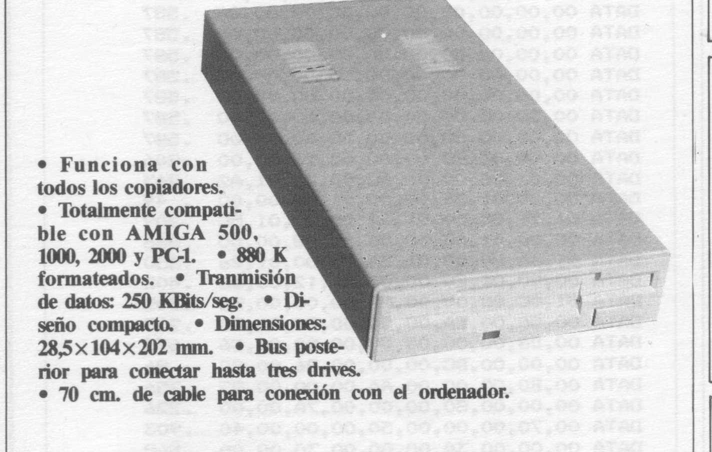
tex-hard s.a.

Corazón de María, 9
Tels. 416 95 62 - 416 96 12
28002 Madrid

MASTER DEALERS
Commodore

Salamanca, 25
Valencia
Teléfono (96) 395 02 43 - Fax 395 02 44

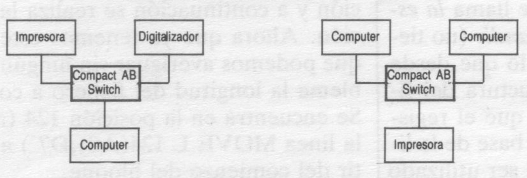
RF 302C SEGUNDO DRIVE DE ALTA CALIDAD PARA AMIGA Y PC-1

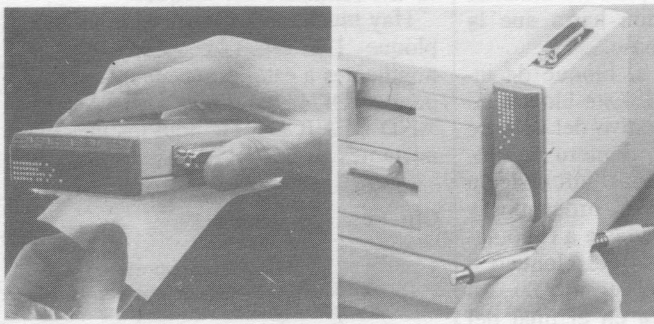


- Funciona con todos los copiadores.
- Totalmente compatible con AMIGA 500, 1000, 2000 y PC-1.
- 880 K formateados.
- Tranmisión de datos: 250 KBits/seg.
- Diseño compacto.
- Dimensiones: 28,5×104×202 mm.
- Bus posterior para conectar hasta tres drives.
- 70 cm. de cable para conexión con el ordenador.

EVITESE REPARACIONES Y MOLESTIAS

MICROSWITCH AB





- Conmutador 1 ordenador, 2 periféricos o viceversa.
- Especial para conectar cualquier DIGITALIZADOR y la IMPRESORA al AMIGA.

P.V.P. **8.500** PTAS.

DIGITALIZADOR DE SONIDO STEREO TH-SOUND



A-500/A-2000/A-1000 • NIVEL DE RUIDO NULO • CONEXION PUERTO PARALELO • COMPATIBLE CON TODO EL SOFTWARE PARA DIGITALIZACION

P.V.P. **12.900** PTAS.

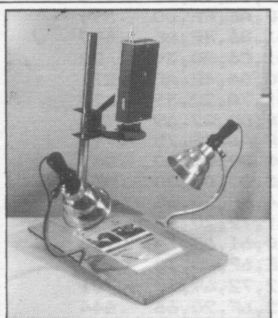
FUNDA AMIGA PARA



1.190 PTAS. PARA A500

1.900 PTAS. PARA A2000

1.190 PTAS. PARA MONITOR



MESAS DIGITALIZADORAS Y CAMARAS

CONDICIONES ESPECIALES PARA DISTRIBUIDORES

Todos los precios no incluyen el I.V.A.

COMMODORE AMIGA

- AMIGA 500 Y 2000
- Ampliaciones de Memoria
- Osciloscopio para AMIGA
- Interface Midi - NOVEDAD
- Digitalizadores video/audio
- GenlockS
- Ratones para AMIGA - NOVEDAD
- Modems
- Tabletas gráficas
- Impresoras NEC

APLICACION COMPLETA AUTOEDICION

- AMIGA 2000
- DISCO DURO 20 MG
- MONITOR MULTISYNC II
- PLACA FLICKER FIXER
- IMPRESORA NEC POSTCRIPT LC 890
- PROFESSIONAL PAGE V1.1

TEX-HARD les ofrece un año de garantía en todos los equipos COMMODORE

```
DATA 20,07,22,3C,00,01,00,02,2C,78  .841
DATA 00,04,4E,AE,FF,3A,2C,79,00,00  .329
DATA 01,70,23,C0,00,00,01,C6,67,00  .223
DATA 00,A8,22,39,00,00,01,BA,24,39  .919
DATA 00,00,01,C6,26,07,4E,AE,FF,D6  .932
DATA BE,80,66,00,00,A4,23,C0,00,00  .151
DATA 01,C2,22,2D,00,04,24,3C,00,00  .316
DATA 03,EE,4E,AE,FF,E2,23,C0,00,00  .816
DATA 01,B6,67,00,00,9A,22,39,00,00  .259
DATA 01,B6,24,39,00,00,01,C6,26,39  .907
DATA 00,00,01,C2,4E,AE,FF,D0,B0,B9  .853
DATA 00,00,01,C2,66,00,00,8E,24,3C  .399
DATA 00,00,02,90,26,3C,00,00,02,AA  . 79
DATA 61,00,00,92,60,00,00,9C,24,3C  .439
DATA 00,00,01,CA,26,3C,00,00,01,ED  . 99
DATA 61,00,00,7E,60,00,00,C0,24,3C  .210
DATA 00,00,01,EE,26,3C,00,00,02,0D  .598
DATA 61,00,00,6A,60,00,00,A2,24,3C  .724
DATA 00,00,02,0E,26,3C,00,00,02,30  .978
DATA 61,00,00,56,60,00,00,84,24,3C  .189
DATA 00,00,02,31,26,3C,00,00,02,4B  .161
DATA 61,00,00,42,60,00,00,70,24,3C  . 71
DATA 00,00,02,4C,26,3C,00,00,02,5D  .399
DATA 61,00,00,2E,60,00,00,42,24,3C  .126
DATA 00,00,02,5E,26,3C,00,00,02,7B  .546
DATA 61,00,00,1A,60,00,00,2E,24,3C  .885
DATA 00,00,02,7C,26,3C,00,00,02,8F  .707
DATA 61,00,00,06,60,00,00,10,96,82  .173
DATA 22,39,00,00,01,64,4E,AE,FF,D0  .399
DATA 4E,75,22,39,00,00,01,B6,4E,AE  .680
DATA FF,DC,22,79,00,00,01,C6,20,39  . 22
DATA 00,00,01,C2,2C,78,00,04,4E,AE  .271
DATA FF,2E,2C,79,00,00,01,70,22,39  . 70
DATA 00,00,01,BA,4E,AE,FF,DC,22,39  .779
DATA 00,00,01,BE,4E,AE,FF,A6,4E,75  .797
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 4E,6F,20,70,75,65,64,6F,20,65  . 42
DATA 6E,63,6F,6E,74,72,61,72,20,66  .884
DATA 69,63,68,65,72,6F,20,61,6E,74  .210
DATA 69,67,75,6F,0A,00,4E,6F,20,70  .269
DATA 75,65,64,6F,20,61,62,72,69,72  .701
DATA 20,61,6E,74,69,67,75,6F,20,66  .644
DATA 69,63,68,65,72,6F,0A,00,4E,6F  .863
DATA 20,70,75,65,64,6F,20,65,78,61  .176
DATA 6D,69,6E,61,72,20,61,6E,74,69  . 40
DATA 67,75,6F,20,66,69,63,68,65,72  .892
DATA 6F,0A,00,4E,6F,20,68,61,79,20  .426
DATA 6D,65,6D,6F,72,69,61,20,64,69  .415
DATA 73,70,6F,6E,69,62,6C,65,0A,00  .249
DATA 45,72,72,6F,72,20,64,65,20,6C  .623
DATA 65,63,74,75,72,61,0A,00,4E,6F  .623
DATA 20,70,75,65,64,6F,20,61,62,72  .518
DATA 69,72,20,6E,75,65,76,6F,20,66  .448
```

```
DATA 69,63,68,65,72,6F,0A,00,45,72  .318
DATA 72,6F,72,20,64,65,20,65,73,63  .643
DATA 72,69,74,75,72,61,0A,00,43,6F  .594
DATA 70,69,61,20,64,65,20,66,69,63  .590
DATA 68,65,72,6F,20,63,6F,6D,70,6C  .941
DATA 65,74,6F,0A,00,00,00,00,00,00  .324
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,00,00,00,00,00,00,00,00  .587
DATA 00,00,03,EC,00,00,00,23,00,00  .206
DATA 00,01,00,00,01,AC,00,00,01,A2  .243
DATA 00,00,01,8E,00,00,01,88,00,00  . 49
DATA 01,7E,00,00,01,62,00,00,01,5C  .209
DATA 00,00,01,4E,00,00,01,48,00,00  .566
DATA 01,3A,00,00,01,34,00,00,01,26  .550
DATA 00,00,01,20,00,00,01,12,00,00  .402
DATA 01,0C,00,00,00,FE,00,00,00,F8  .305
DATA 00,00,00,EA,00,00,00,E4,00,00  .209
DATA 00,D6,00,00,00,D0,00,00,00,C6  .673
DATA 00,00,00,BC,00,00,00,B6,00,00  . 96
DATA 00,B0,00,00,00,A6,00,00,00,92  .756
DATA 00,00,00,80,00,00,00,7A,00,00  .226
DATA 00,70,00,00,00,50,00,00,00,40  .903
DATA 00,00,00,3A,00,00,00,30,00,00  .568
DATA 00,1A,00,00,00,05,00,00,00,00  .421
DATA 00,00,01,72,00,00,00,02,00,00  . 83
DATA 00,6A,00,00,01,9C,00,00,00,08  .764
DATA 00,00,00,00,00,00,03,F2        . 88

Numero de lineas: 208
```

Listado 7 (continuación)

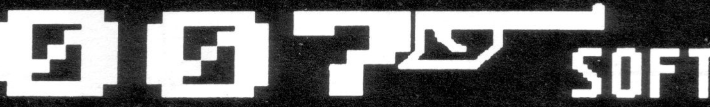
del nombre del fichero a copiar y lo obtenemos como ya hemos dicho de la dirección de la etiqueta paramadd. A continuación el modo de acceso (lectura = READ = -2) y se efectúa la función teniendo siempre cuidado de apuntar en A6 la base de la librería con «—DOSBase». El registro D0 que se devuelve al ejecutarse la función es el que proporciona la dirección de la estructura de ese bloqueo. Por supuesto lo salvamos a «RLOCK» para su posterior utilización en otras funciones.

El segundo paso es abrir el fichero a copiar o ANTIGUO FICHERO. Dos parámetros o registros se necesitan para realizarlo: uno apunta al nombre del fichero como en la anterior función y el segundo el modo de acceso (ANTIGUO FICHERO = 1005) y se llama a la función. La dirección devuelta en el registro D0 es lo que se llama *la estructura del fichero* o *filehandle* (no tiene traducción exacta por lo que desde ahora la llamaremos estructura del fichero). También observa que el registro A6, el que contiene la base de la librería, queda intacto para ser utilizado en la siguiente función hasta que la cambiemos por otra base.

Ahora obtenemos el bloque de información del fichero. Este bloque informa al sistema operativo del nombre del fichero, longitud, número de bloques en el disco, y más información muy importante para el ordenador y para nosotros. Para llevarse a cabo necesitamos la estructura del bloqueo o «lock» (RLOCK) en el registro D1, así como 262 bytes (mira en el final del programa) alineados a larga palabra (LONG WORD) que será el buffer para almacenar el bloque de información y a continuación se realiza la función. Ahora que ya tenemos este bloque podemos averiguar sin ningún problema la longitud del fichero a copiar. Se encuentra en la posición 124 (mirar la linea MOVE.L 124(A0),D7) a partir del comienzo del bloque.

Hay mucha más información en este bloque. Los offsets o valores correspondientes a éstos se encuentran en la página D-74 del libro LIBRARIES AND DEVICES. He aquí los más importantes:

Offset dec.	Información
4	Tipo de directorio. Si es menor que cero no lo es.
8	Nombre del fichero, terminado por un cero.
116	Bits máscara de protección: READ=3



ALCALA, 211.
28028 MADRID.
Tel. 91/256 14 30.
Fax 91/256 16 91

DISTRIBUIDOR OFICIAL COMMODORE



PUNTO OFICIAL DE VENTA DE AMIGAZETA

Tratamientos de textos

- Wordperfect - 48.000
- Textcraft Plus - 9.500
- Prowrite - 20.000
- Textpro - 14.000
- Kind Words (Español) - 13.500
- Calligrapher - 23.000
- Ashas.Calligra.Fonts - 13.500
- Call. Newsletters Fonts - 8.000
- Call. Studio Fonts - 8.000
- Fancy 3D Fonts - 12.000
- Zuma Fonts Vol. 1.2.3.4 - 7.000
- Aloha Fonst Vol. 1.2.3 - 6.500
- NTF Fonts Vol. 1.2.3 - 14.000
- Aloha Fonts (del 1 al 3) - 6.500

Paquete Commodore - 30.000
(Superbase,Logistix,Musica.......)

Amiga Hardware

- Expans. 8M A2000 -395.000
- Expans. 2.M A2000(Ampl.8 M) -98.000
- Expans. 1/2 M - 33.000
- Expans. 2M A500 -140.000
- Modulador T.V A-500 - 6.500
- Placa Video Comp. A-2000 -15.000
- Placa XT +Unidad 5 1/4 - 95.000
- Placa AT (1 Mega de memoria) -185.000
- Digitalizador Stereo Audio - 20.000
- Perfect Sound Digit. - 25.000
- Prosound Designer Digit. - 25.000
- Future Sound Digitalizador - 32.000
- Digitot (Audio + Video) - 34.500
- Funda Polivinilo A-2000 - 2.500
- Funda Polivinilo A-500 - 1.975
- Archivador 10 diskettes - 300
- Archivador Metalico 70 D - 4.000
- Archivador Plastico 150 D - 3.500
- Alfombrilla Para Raton - 2.000
- Conmutador 2 perif. paralelo - 16.500
- Impresora Color Nec 24 Agujas P6 Plus 80 C./264 C.P.S. -170.000
- Impr. Color Star 9 LC-10 - 75.000
- Tableta Grafica Easyl A-2000 - 90.000
- Tabl. Grafica Easyl A500/A1000 - 80.000
- Genlock Rendall/Ariadne - 79.000
- Genlock con funcion Invert - 70.000
- Genlock Commodore A-2000 - 49.000
- MiniGen (Genlock) - 39.000
- DigiView Gold 3.0 PAL - 25.500
- Adaptador D.View A500/A2000 - 5.000
- Digi Droid - 15.500
- Digipic(Tiempo real monoc.) - 70.000
- Perfect Vision(T. real monoc.) - 49.000
- Unidad 3 1/2" Ext Commodore - 35.000
- Unidad 3 1/2" Ext No Commod. - 30.000
- Unidad 3 1/2" Int Commodore - 30.000
- Unidad 3 1/2" Int No Commod - 25.000
- Unidad 5 1/4" Ext (80 pistas) - 40.000
- Unidad 5 1/4" Ext (40 pistas) - 35.000
- Disco Duro 40 Megas A-2000 -185.000
- Disco Duro 20 Megas A-2000 -115.000
- Disco Duro 20 Megas A-500 -140.000
- Midi 2 Out - 24.000
- **Leedor De Pistas** - 20.000
- **Proccesor Accelerator** - 35.000
- Flicker Master (Filtro) - 4.500
- Euroconector Amiga-Sony Trin. - 5.500
- Conector RGB 23 pines - 1.500
- **Separador Color** - 60.000
- **Frame Buffer 2 Megas** 130.000
- **Genlock** - 88.500
- **Disco D. 20 M A-500 Comm-98.000**

Amiga librería

- Amiga Para Principiantes - 4.134
- 68.000 Guia Del Usuario - 1.900
- First Steps In Assembly Lang - 5.475
- Alphabets - 3.500
- Computer Animations - 5.475
- Amiga Applications - 4.675
- Amiga Handbook - 4.775
- Advanced Amiga Basic - 5.975
- Am. System Programmer's Guide - 5.475
- ROM Kernel Ref.Manual:EXEC. - 5.475
- ROM K. Ref.Manual:Libraries - 7.975
- Developers Reference Guide - 3.500
- A.System Programmer's Guide - 5.975
- C: Progr. Graph. On Amiga And Atari ST - 4.500
- Inside Amiga Graphics - 4.900
- Compute's A.Dos Réf. Guide - 4.900
- 68000 Assembly Language - 5.975
- Amiga Machine Language - 5.475
- Bantam's Amiga Dos Manual - 5.975
- Inside The Amiga With C - 5.975
- Hardware Reference Manual - 5.475
- Becomming an Amiga artist - 5.475
- Amiga for Beguinners - 5.475
- Amiga Basic-Inside and Out - 5.595
- Amiga 3D Graphic in Basic - 5.475
- Amiga Tricks and Tips - 5.475
- Amiga Dos: Inside and Out - 5.475
- Amiga Disk Drives - 5.995
- Amiga C for Beginners - 5.975
- Using Deluxe Paint II - 5.475

Revistas

- Amiga World(USA) - 830

Autoedición

- Pagesetter - 25.000
- Professional Page - 60.000
- **PAGESTREAM** - 35.000

Base de Datos

- Superbase Personal - 19.500
- Video Wizard - 12.500
- Data Retrieve - 14.000
- **MICROFICHE FILER PLUS** - 25.000

Comunicaciones

- Aegis Diga! - 13.500

Emuladores

- Dos 2 Dos - 12.500
- Disk 2 Disk - 12.500
- Transformer (pc) - 6500
- 64 Emulator II - 12.000

Gestión

- Analyze 2.0 - 25.000
- Quarterback - 11.000
- Galileo 2.0 - 17.000
- **EZ-BACKUP** - 11.000
- **SUPERBACK** - 11.000

Gráficos / Vídeo

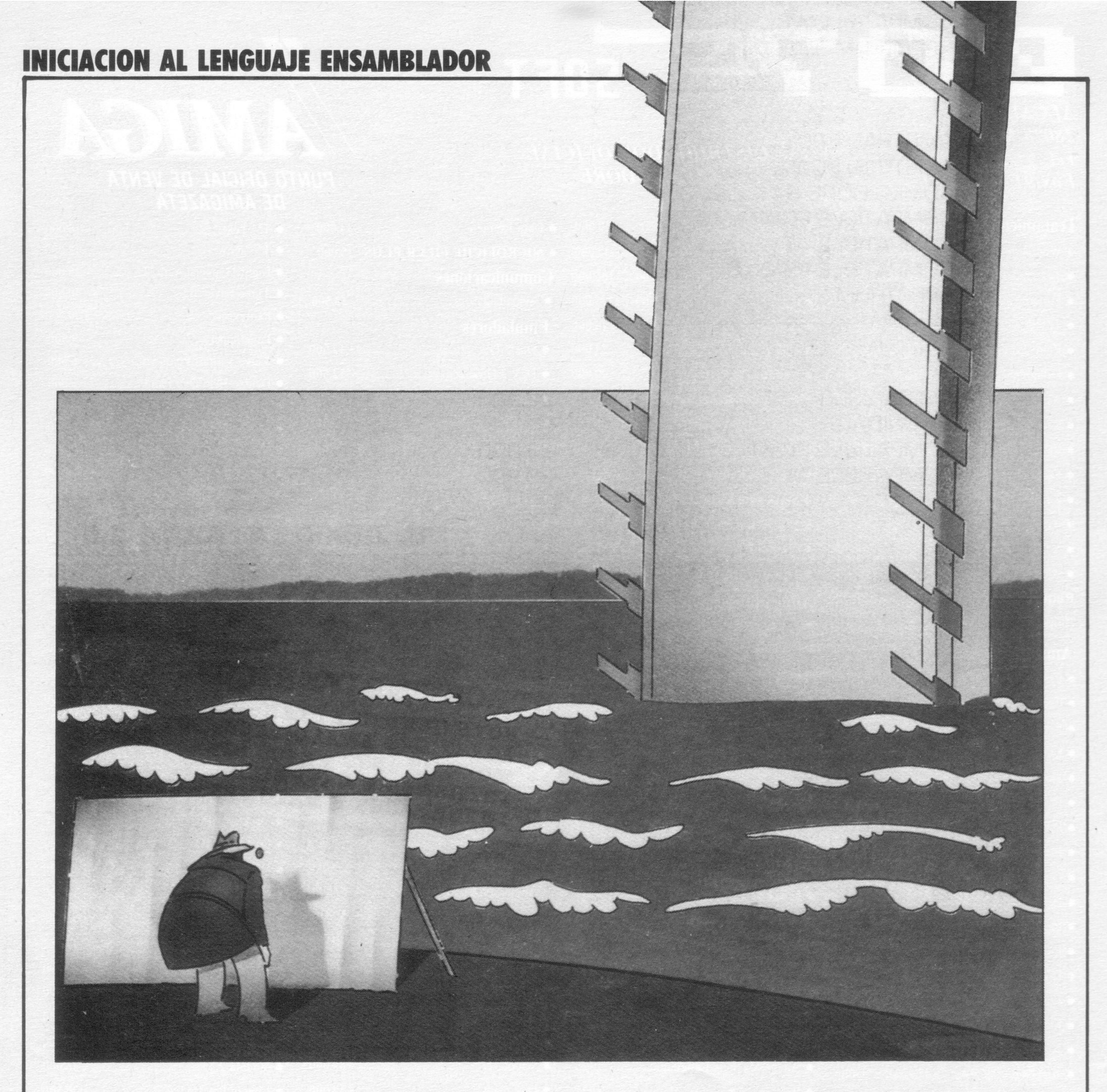
- Printm. Plus Art Gallery 1 - 7.000
- Printm.Plus Art Gallery 2 - 7.000
- Aegis Videoscape 3D - 30.000
- Aegis Videoscape 3D 2.0 - 35.000
- Aegis Impact - 14.000
- Aegis Animator+Aegis Images - 22.000
- Animation Apprentice - 40.000
- Animation Multiplane - 13.000
- Animation Editor - 10.000
- Animation Effects - 9.000
- Animation Stand - 9.000
- Animation Flipper - 9.000
- Pro Video CGI Plus Pal - 56.500
- Pro Video Set Fonts I - 20.000
- Pro Video Set Fonts II - 20.000
- Deluxe Library - 5.000
- Sculpt 3D - 25.000
- Animate 3D - 25.000
- Sculpt 4D - 85.000
- Digi paint - 12.000
- Interchange 3D - 10.500
- Int Forms In Flight Conv. mod. - 5.000
- Interchange Objects Disk 1 - 4.000
- **INTERFONT** - 18.000
- Introcad 2.0 - 12.000
- New Technology Coloring Book - 7.000
- Deluxe Video 1.0 - 10.000
- Deluxe Video 1.2 - 13.500
- Aegis Videotitler - 25.000
- Aegis Modeler 3D - 16.000
- **PHOTON VIDEO C.ANIM** - 35.000
- Graphicraft - 6.500
- Dynamic Cad 2.3 - 65.000
- Forms In Flight II - 16.000
- Express Paint - 18.000
- Deluxe Productions - 39.000
- Deluxe Photolab - 13.500
- Deluxe Paint II Pal - 13.500
- **DELUXE PAINT III** - 25.000
- **PHOTON PAINT 2** - 25.000
- TV Text - 18.000
- TV Show - 18.000
- Lights!,Camera!,Action! - 25.000
- Turbo Silver 3D - 40.000
- Fantavision - 20.000
- Pageflipper - 9.500
- Video Visions Fonts vol.1 y 2 - 6.500
- Deluxe Help Digi Paint - 6.000
- Deluxe Help Photon Paint - 6.000
- Deluxe Help Deluxe II - 6.000
- Video Effects - 45.000
- Invision - 25.000
- Clip Art Magnetics 1 al 8 - 5.000
- Cad Parts - 6.000
- Fine Print - 8.500
- Turbo Print - 8.000
- Desktop Artist 1 - 6.500
- Comic Setter - 14.000
- Comic Setter Science - 7.000
- Comic Setter Heroes - 7.000
- Comic Setter Figures - 7.000
- **ZOETROPE** - 23.500
- Human Design - 7.500
- Macrobot D. - 7.500
- Future D. - 7.500
- Architectural D. - 7.500
- Desktop Artist - 6.500
- **DESIGN 3D** - 18.000
- Moviesetter - 18.000
- **C-LIGHT** - 18.000
- Cad Parts - 6.000

Lenguajes

- Assempro - 18.000
- Lattice DBC III Library - 25.000
- Aztec C - 38.000
- Aztec C SDB(for DBase) - 15.000
- Modula 2 - 38.000
- Library For Modula 2 - 29.500
- Arexx - 9.000
- Metacomco Pascal - 19.000
- Pro Fortran 77 - 40.000
- Multi Forth - 19.000
- Devpac - 18.000
- MacroAssembler - 20.000

Música

- Aegis Sonix 2.0 - 15.000
- Aegis Audiomaster II - 18.000
- Deluxe Music Construction Set - 13.500
- The Music Studio - 13.000
- Pro Sound Designer - 9.000
- Pro Sound Midi Plus - 10.500
- Dr.T's KCS 1.6A - 45.000
- Dr.T's Copyist Professional - 55.000
- Midi Magic - 22.000
- E.C.T. Sampleware (del 1 al 4) - 6.500
- Dynamic Drums - 9.500
- A Drum - 9.500
- **SOUND OASIS** - 14.000



WRITE=2
EXECUTE=1
DELETE=0.

124	Longitud del fichero.
128	Número de bloques.
132	Fecha del último cambio.
144	Comentario, terminado en cero.

Por lo tanto para obtener por ejemplo el número de bloques del fichero tenemos que leer la posición 124 a partir del comienzo del bloque de información que nos ha dado la función EXAMINE. Queda en tu elección el darle utilidad al resto de los offsets.

Volviendo al programa, una vez conocida la longitud total del fichero hay que reservar un espacio igual de grande que el fichero, y para ello usaremos las función ALLOCMEM. Sólo serán necesarios dos parámetros, el número de bytes a reservar u ocupar que es igual a la longitud del fichero y el tipo de memoria que en este caso hemos elegido: MEMORIA CHIP (=2) y MEMORIA BORRADA (=10000). Se suman los dos valores (=10002) y lo insertaremos en el registro D1.

Hemos usado MEMORIA CHIP pensando en muchos usuarios que disponen sólo de 512 k, pero si tienes ampliación te recomiendo utilizar MEMORIA FAST (=4, valor definitivo= 10004). El valor devuelto en el registro D0 después de la función será el lugar en donde se encuentra el bloque de memoria, que salvaremos en algún puntero previamente insertado al final del programa, en este caso MEMBLK.

Sólo queda leer el contenido del fichero en el bloque de memoria, mediante la función READ, para ello se necesitan tres parámetros. En D1 colocaremos la estructura del fichero (filehandle) que nos dio la función OPEN, en D2 cargaremos la posición del bloque de la memoria y en D3 la longitud total del fichero.

Una vez leído todo el fichero tendremos que abrir el nuevo fichero que contendrá los datos leídos por el antiguo. El proceso se acorta pues es suficiente abrirlo (modo acceso = 1006) y escribir sobre él apuntando el bloque de memoria sobre la función WRITE. Si has comprendido esto lo demás es sencillo. Todo lo que hayamos abierto hay que cerrarlo, desocupar el bloque de memoria, cerrar los dos ficheros y desbloquear el fichero fuente. Es muy importante que lleves un orden, pues si el programa se va a mejorar no sa-

brás qué has abierto o qué has cerrado.

Observa también que si ocurre algún error dentro del programa te envía mensajes de error. Mira el proceso que sigue hasta que lo imprime en pantalla. Tan sólo es necesaria la función WRITE, apuntando al texto a imprimir, dándole la longitud del texto (observa la resta entre las etiquetas), y apuntando a la salida estándar de mensajes CLI que ha sido previamente abierto con el programa NEWASTARTUP.

Como dijimos al comienzo el AmigaDOS te permitía copiar cualquier fichero a cualquier periférico del ordenador. El cambio es sencillísimo, simplemente el fichero destino debe ser PRT:, u otro cualquier periférico. Ejemplos:

prg fichero PRT:	Impresora.
prg fichero SER:	Modem.
prg fichero CON:0/20/640/150/	Este abre una ventana por donde saldrá el fichero.

o cualquier periférico que hayas abierto. Estos son los cambios en el programa para el último caso (ver listado 4):

MOVE.L 4(A5),D1 apunta al nombre del nuevo fichero

por

MOVE.L #NEW-F,D1 apunta al nombre del nuevo fichero

e insertar en el bloque de mensajes la línea

NEW–F DC.B «CON:0/20/640/150/FICHERO»,0

Ahora el programa funciona con un solo parámetro, pues el otro ya lo hemos insertado en el programa.

Detección de errores

Veamos una función también importante, IoErr(), que permite conocer con profundidad el error que se ha cometido al abrir o utilizar un fichero o directorio. Se utiliza después de las funciones como OPEN, WRITE, CLOSE, etcétera. La llamada a IoErr() no tiene ningún parámetro y el resultado devuelto en el registro D0 junto con la tabla 2 permite conocer el error exacto.

Como bien observarás, esta tabla corresponde con el set de errores del CLI que está también en el libro que acompaña al ordenador (página B7).

Posicionamiento en el fichero

Por último veamos un poco la función SEEK. Esta nos permite situar nuestro supuesto cursor en el fichero para que comencemos a leer o escribir sobre el fichero. vamos a ver en la figura diferentes situaciones:

- Para situarnos en el comienzo del fichero esta sería la rutina:
- Para situarnos al final del fichero:
- Para situarnos 100 bytes del comienzo del fichero:

Recuerda que en esta función el registro D2 especifica el número de bytes en adelantar o atrasar y que el registro D3 es el modo de hacerlo (PRINCIPIO, FINAL, ACTUAL). Esta función se ha de utilizar cada vez que queramos acceder a cualquier parte del fichero y antes de la función WRITE. Esto es todo en este segundo capítulo; en el siguiente capítulo veremos cómo se accede en profundidad a los periféricos y también las más importantes funciones de que se compone la librería Exec. ■

Dominio Público

Esta sección está dedicada a breves comentarios de aquellos programas de dominio público que puedan ser de especial interés para todos los usuarios de Commodore.



Existe una serie de programas de dominio público que pueden calificarse de «un solo uso», como programas específicos de juegos, contabilidad, bases de datos, etc. Y hay una amplia gama de pequeños programas, lo que se conoce con el nombre de «utilitarios», cuya aplicación para un fin concreto no está nada clara, si bien es cierto que cualquier usuario puede aprovecharse de ellos.

Algunos de estos programas son pequeños «añadidos» a programas de tipo comercial que, por cualquier razón, no disponen de ello. Por ejemplo: ¿Por qué junto con Deluxe Paint no se suministra un visualizador de pantallas o series de pantallas? ¿Cómo es que el Sonix no dispone de un «player» para ejecutar una canción?

Recientemente ha llegado hasta nuestras manos un disco llamado **ART #1**, que contiene todo tipo de pequeños programas utilitarios, especialmente orientados hacia el mundo de los gráficos y, sobre todo, los iconos del Amiga. Muchos usuarios encontrarán en ellos los programas que han estado buscando durante mucho tiempo.

AltIcon es un utilitario que permite obtener el conocido efecto de «mini-animación» en los iconos, mezclando dos imágenes distintas de un mismo icono en uno sólo. El primer dibujo muestra el icono normal, mientras que el segundo sólo aparece cuando se hace un click sobre él.

Brush2C es lo que muchos programadores han estado buscando desde hace bastante tiempo. Permite obtener, partiendo de un brush IFF (como los que se graban con Deluxe Paint o Aegis Draw), un listado de estructura en C, de modo que puede ser utilizado mediante #include al principio del programa, como un fichero «.h». La imagen puede pasar también con toda la información de los registros de color.

Una amplia serie de pequeños pero prácticos programas relacionados con los gráficos del Amiga.

Brush2Icon es el nombre que aquí ha tomado el también práctico «ZapIcon». Permite convertir un brush IFF en un icono tipo «tool» de cualquier tamaño. Para la edición del icono se recomienda utilizar Deluxe Paint en una resolución de 640x200, para evitar posteriores deformaciones por culpa de la resolución. Además, también es conveniente tener en cuenta que los colores por defecto de DPaint (negro, marrón claro, azul y naranja) no se corresponden con los colores por defecto del Workbench (azul, blanco, negro y naranja). Para el cambio es conveniente modificar la paleta de DPaint.

DPSlide es uno de los más conocidos programas para preparar Slide-Shows. Un Slide-Show es una «presentación» de pantallas gráficas, que van apareciendo una por una en el monitor. Siempre existe la posibilidad de variar la velocidad y la forma en que aparecen: fundidos a negro, scroll vertical, instantánea... DPSlide admite pantallas en todas las resoluciones y de todos los colores. Este es el programa de este tipo más comúnmente utilizado por todos los aficionados al dominio público.

HamEditor es un pequeño programa de dibujo en HAM que puede ser utilizado para retocar las pantallas obtenidas con programas como Digi-View, si no se dispone de nada mejor. Aunque limitado, puede ser útil a la hora de escribir algo de texto o hacer retoques.

IFFdump puede ayudar a los programadores que desean comprender mejor la estructura de los ficheros IFF. Cuando a IFFDump se le indica el nombre de un fichero IFF, lo descompone y muestra por pantalla todas sus características.

IFFencode es seguramente uno de los más interesantes utilitarios de este disco. Al igual que otros programas ya conocidos, como **Grabbit!** o **ScreenX**, permite «capturar» cualquier pantalla que aparezca en el monitor del Amiga y grabarla en un fichero IFF del disco. Esta pantalla puede posteriormente retocarse con cualquier paquete de dibujo de los muchos que existen.

Lens resulta más curioso que práctico. Al arrancarlo, una pequeña ventana hace de «lente de ampliación» de la zona en la que se mueve el puntero del ratón.

MacView es uno de esos extraños «conversores» de gráficos. Permite visualizar en el Amiga pantallas de Macintosh grabadas con el programa MacPaint. A pesar de las diferencias de formato gráfico (los pixels del Mac son cuadrados) la pantalla puede verse en formato 320x200 ó 640x400, los más similares.



Printscrn es un sustituto del conocido GraphicDump que se suministra con el disco del Workbench. Permite obtener volcados por impresora de cualquier pantalla gráfica del Amiga, pudiéndose seleccionar a voluntad.

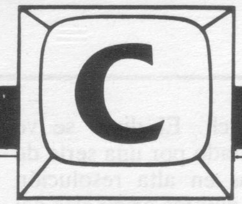
Popcolors demuestra la potencia de algunos diminutos programas para «inmiscuirse» en otros programas ajenos. En este caso, Popcolors permite modificar en cualquier momento la paleta de colores de cualquiera de los «screens» o pantallas situadas en la memoria del Amiga. El cambio de colores afecta, naturalmente, a todo gráfico situado en la pantalla en cuestión.

Entre los programas destinados a servir como projectores de pantalla están **SeeILBM, Show** (el más útil de todos, sin duda) y **ShowHAM.** Un buen equipo de utilitarios gráficos de pantallas para todos los gustos.

Por último, también están incluidos **Xicon** y **Ticon,** dos pequeños programas diseñados para permitir la ejecución de programas y ficheros «batch» desde el Workbench. El disco se ve complementado por una serie de 13 pantallas en alta resolución basadas en dibujos originales del conocido Hagen.

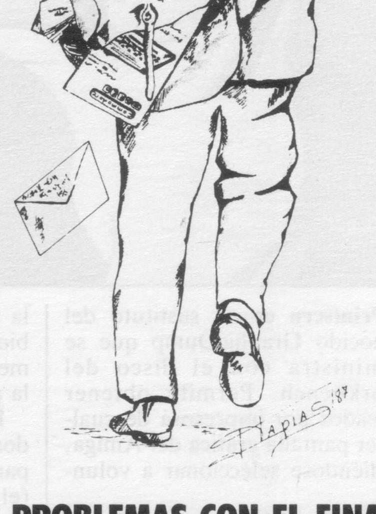
Seguro que entre estos pequeños programas aquí reseñados se encuentra alguno de los que has ido buscando durante largo tiempo. Y para terminar, una pregunta como las que muchas personas se han hecho tras la aparición del Workbench 1.3: ¿Por qué Commodore no se decide de una vez por todas a incluir programas de este tipo en sus discos del sistema o de utilitarios? En el Workbench 1.3 ya hemos podido observar la entrada de **Less**, el visualizador de ficheros de texto y de otros «simpáticos» programillas, como el reloj en el puntero y similares. Esperemos que dentro de poco se suministren algunos de estos programas en la configuración básica de cualquier Amiga. ■

Los programas comentados en esta sección son Dominio Público. Pueden conseguirse copias de ellos a través de amigos, grupos de usuarios, clubs, BBSs o compañías especializadas.



ARTAS DEL LECTOR

Esta sección es un buzón abierto a vuestras consultas, comentarios y sugerencias.



DUDAS AMIGABLES

Poseo un C-64 y estoy pensando en cambiarlo por un Amiga 500. No obstante tengo algunas dudas:

1. Con el Transformer ¿Podré ejecutar programas de PC, como por ejemplo Hojas de Cálculo (Lotus 1-2-3), Bases de Datos (dBase II) o procesadores de texto?

2. Si conecto el Amiga a un televisor de 14″ con el emulador, ¿obtendré una resolución suficiente como para trabajar con los programas mencionados?

3. Si se conecta un disco de 5 1/4 al Amiga, ¿Se puede intercambiar información en modo PC con el de 3,5?

4. Finalmente, ¿Tienen noticias sobre si Commodore va a reducir el precio del Amiga 500 en un futuro próximo?, teniendo en cuenta que en Inglaterra cuenta unas 20.000 Pts. menos.

Oriol Borrell Ribot
Barcelona

1. Aunque realmente sí se pueden ejecutar todos estos programas con el Transformer, su extremada lentitud (menos de la mitad de velocidad de un PC XT clásico) no los hace muy apropiados para trabajar con el Amiga. Lo más recomendable es transferir los datos ya existentes al Amiga y aprovecharlos con otros programas similares, como *Maxiplan 500, SuperBase* o *WordPerfect,* que en algunos casos son mejores que los que tú citas.

2. La resolución de un televisor de 14″ deja bastante que desear, aunque puede trabajarse con los programas que tú mencionas. Lo más recomendable, sin embargo, es trabajar en modo 60 columnas en vez de 80.

3. Para intercambiar datos entre discos de 3,5 Amiga-PC hace falta un programa especial, por ejemplo *DOS-2-DOS*.

4. Todos los asuntos sobre bajadas de precio suelen ser «confidenciales», pero de momento no tenemos ninguna noticia al respecto. Respecto a lo que comentas del precio en Inglaterra, no andas desencaminado: El precio actual de un Amiga 500 con monitor 1084S es de 599 libras (120.000 ptas), en EE.UU. son 799 dólares (96.000 ptas), mientras que aquí son 183.000 ptas (105+59+IVA)... ¡sólo el doble!

CARACTERES CASTELLANOS CON EASY SCRIPT

Tengo un C-64, MPS 801 y el procesador de textos **Easy Script**. Quisiera saber cómo hay que hacer para asignar caracteres redefinidos (vocales acentuadas, eñes, etc.) a diferentes teclas, ya que al pulsar F1/$, como dice el manual, sólo se imprime el último carácter, ch xx...xx, redefinido. El n.º 3 de CW no me ha aclarado las ideas.

Jorge Abellán Vázquez
Hospitalet del Infante, Tarragona

Poca más información podrás encontrar aparte de la que se apunta en el manual y el artículo del número 3, pero es más que suficiente para aprovechar todas las posibilidades de la impresora. Para el caso de la MPS-801, se pueden definir hasta 10 caracteres diferentes. La instrucción que hay que teclear es:

<F3>n=ch:sa x,x,x,x,x,x,x,x...

El <F3> indica que se debe pulsar la tecla F3, que en la pantalla aparece como un asterisco en vídeo inverso. La «n» corresponde al número de carácter (0 a 9) que se desea redefinir. «CH» es el código ASCII al que va a asignarse. «SA» indica la dirección secundaria que se enviará a continuación del carácter definido y «x,x,x,x...» la serie de códigos en cuestión. Ejemplo: *0=254:sa5,0,95,72,68,66,65,0,0. Hay que tener en cuenta que esta línea es sólo la *definición* del carácter especial. Para utilizar el carácter en cuestión, hay que pulsar la tecla F1 seguida del número de carácter definido. En este ejemplo, sería <F1> y 0.

Como en la MPS-801 hay que redefinir los caracteres que se quieren crear, ya que no se pueden utilizar «trucos» como el carácter de retroceso (backspace), como sucede en otras impresoras. Este proceso se comenta ampliamente en el manual de la impresora.

PROBLEMAS CON EL FINAL CARTRIDGE III

1. Cuando utilizando el Final Cartridge III, congelo un juego, y hago un volcado de pantalla, ¿Cómo se puede volver de nuevo al juego, como pasaba en el FC II?

2. Tengo también el FC II pero resulta que desde hace una temporada, cuando quiero sacar un volcado de pantalla, la impresora no imprime, aunque no es culpa de la impresora, sino del cartucho, ya que con el FC III funciona perfectamente. ¿A qué puede ser debido esto? ¿Qué debo hacer si es una anomalía del cartucho?

3. Quisiera saber si existe alguna otra forma para conectar una impresora Centronics al CBM-64, que no sea con el port del usuario y con el FC enchufado.

4. También quisiera saber cómo se mandan en Basic los códigos para la impresora, ya que con el GWBasic del PC, me funciona bien, un LPRINT CHR$(27)+CHR$(50), donde 27 y 50 son los códigos que he de mandar, pero esto no funciona en el Commodore, por supuesto, abriendo el canal para la impresora, y utilizando PRINT#1,CHR$(27)+CHR$(50). Pues bien, no se cómo hacerlo.

5. Hay veces que, por equivocación, se crea en el disco, un fichero secuencial que tiene el nombre «,», es decir, que sale así en el directorio:

1 «, » SEQ

Envía tus cartas a: Commodore World, Rafael Calvo, 18, 4.° B, 28010 Madrid.

Quisiera saber cómo eliminar estos ficheros, ya que el comando DOS «S:,» no funciona.

6. Una cosa que llevamos observando varias personas, es que el cartucho FC, produce algo en el Commodore que hace que éste se quede semicolgado por unos instantes, aunque estos instantes sean años, pero después vuelve a funcionar con toda normalidad. ¿Por qué pasa esto? ¿Cómo se puede evitar?

7. ¿En qué número de vuestra revista, si es que lo han publicado, puedo encontrar un programa, el cual lea el directorio tal y como sale cuando tecleamos DOS«$, y no sólo leyendo los nombres de los ficheros?

8. Con el Amiga-500, trabajando con un emulador C-64, ¿Se puede trabajar con la unidad de 3,5 en modo Commodore-64? ¿Puede la 1541 actuar en modo PC?

Javier Iglesias Madrigal
Valladolid

1. Dependiendo del juego, a veces es posible pero otras veces no. El único sistema cuando el juego no quiere seguir es volver a cargarlo.

2. Puede que no sea una anomalía del cartucho, sino que simplemente no funcione con algún juego en concreto (dado que el FC II es algo antiguo). Sin embargo, si compruebas que el cartucho finalmente está estropeado, tendrás que dirigirte al sitio en el que lo compraste para que te lo reemplacen por uno en condiciones.

3. Existe una serie de interfaces por hardware que se conectan en el port serie del C-64 (donde la unidad de discos y las impresoras Commodore), que permiten que la impresora Centronics se comporte como una impresora serie normal. También puede utilizarse el cable en el port del usuario y un programa de código máquina para hacer la conversión.

4. El sistema para enviar códigos ASCII a la impresora es prácticamente el mismo. La secuencia en cuestión sería: OPEN 1,4:PRINT#1, CHR$(27); CHR$(50).

5. Si el fichero se llama «,», es decir, una coma sin más espacios, puede borrarse con OPEN1,8,15: PRINT#15,«S0:?», instrucción que borra todos los ficheros de una sola letra (¡mucho cuidado!) Si lleva más espacios, habrá que añadir éstos a continuación de la interrogación. El equivalente en el FC es: DOS«S:?».

6. Es la primera noticia que tenemos de ello. El único problema que nosotros hemos observado, en el FC II, es que al imprimir con la Riteman C+ puede quedarse bloqueado y dejar de imprimir.

7. Te puede servir el programa «DIR» que va incluido en el disco TEST/DEMO de la 1541 (también en el manual de la unidad). Nosotros también hemos publicado algún programa con rutinas de este tipo, como Datafile (16-17-18), Doctor de Discos (19), Comando DIR (28), Ordena tus discos (21-24-26).

8. Aunque el C-64 sí puede aprovecharse de la unidad de 3 del Amiga, teniendo en cuenta que no es compatible con la 1581, la 1541 no puede utilizarse en modo PC, por problemas evidentes de capacidad y formato.

DISCOS, PROGRAMAS Y EPROMS

Poseo un C-64 y quisiera saber: ¿Qué es el CP/M?, ¿Dónde podría conseguir el cartucho Z80? ¿Poseo la unidad de discos 1541 II y no se dónde conseguir programas para mi unidad? ¿Para qué sirven las tarjetas Eprom?

David Rosado Núñez
Barcelona

El CP/M es un sistema operativo que gozó de cierta popularidad hace unos cuantos años y que actualmente está en desuso, por culpa del MS-DOS. Una de sus principales características era la compatibilidad entre distintos ordenadores, como sucede con los PCs. Para que el C-64 trabaje en este sistema es necesario el cartucho que incorpora el microprocesador Z80, que nunca ha llegado a distribuirse oficialmente en nuestro país.

Existe una gran cantidad de programas para la 1541, entre aplicaciones y juegos. Para comprarlos o conseguir listas de productos te recomendamos que te pongas en contacto con los diferentes distribuidores que se anuncian en las páginas de nuestra revista.

Las tarjetas Eprom son tarjetas en las que va instalada una memoria Eprom, un tipo especial de memoria grabable pero no borrable (excepto con técnicas especiales) que se asemeja mucho a una ROM. Las hay de distinta capacidad, desde pocos bits hasta 8, 16, 32, 64, 128K o más. Sirven principalmente para que el usuario pueda crearse sus propios cartuchos con programas y aplicaciones que arrancan instantáneamente y a las que se puede acceder al instante. Para grabarlas es necesario un aparato grabador de Eproms y un programa espacial, y para borrarlas un aparato borrador de Eproms.

AMIGA-TORTUGA

Os escribo para preguntaros sobre una duda con el Amiga. Hace poco he intentado hacer muchas cosas en C, pero siempre me he encontrado con una gran lentitud. A la hora de mover BOBs en C, tengo que hacerlos con 4 colores como mucho, porque si no no es posible obtener una buena velocidad. Cuando quiero dibujar Pixels individuales, me ocurre lo mismo. Teniendo en cuenta que son 7 Mhz, que es un Amiga y está en C compilado, ¿Qué es lo que ocurre? Incluso el Commodore-64 es más rápido con su código máquina que el Amiga.

Javier Guerrero Díaz
Algeciras, Cádiz

Aunque todos los lenguajes tienen sus limitaciones de velocidad, no es éste el caso del C en el Amiga. Al ser un lenguaje compilado y además el lenguaje nativo del ordenador, su velocidad es verdaderamente alta. Pero también es cierto que el Ensamblador es mucho más rápido. Puede que tu problema resida en no estar utilizando las librerías y funciones óptimas para el movimiento de objetos, o que la memoria que empleas para los gráficos no sea del tipo adecuado. Pero como podrás observar, la mayoría de los programas de dibujo y muchos juegos están escritos en C (Deluxe Paint II, por ejemplo) y no puede decirse que sean lentos precisamente.



DOS-2-DOS

Ordenador: Amiga
Distribuidor: Sin distribuidor oficial

Hay muchas personas que opinan que en el Amiga no tendría por qué existir ningún tipo de compatibilidad con otros modelos de microordenadores y ordenadores personales. Pese a ello, existe un gran número de buenas razones para proporcionar compatibilidad al Amiga. En esta revista ya se han comentado en ocasiones productos como los conocidos 64 Emulator y Go 64!, que permiten acceder al mundillo del Commodore 64. En esta ocasión, DOS-2-DOS es un conversor de formatos de disco Amiga-PC, no un emulador, es decir: permite convertir ficheros pero no ejecutar los programas. Esto es importante tenerlo en cuenta.

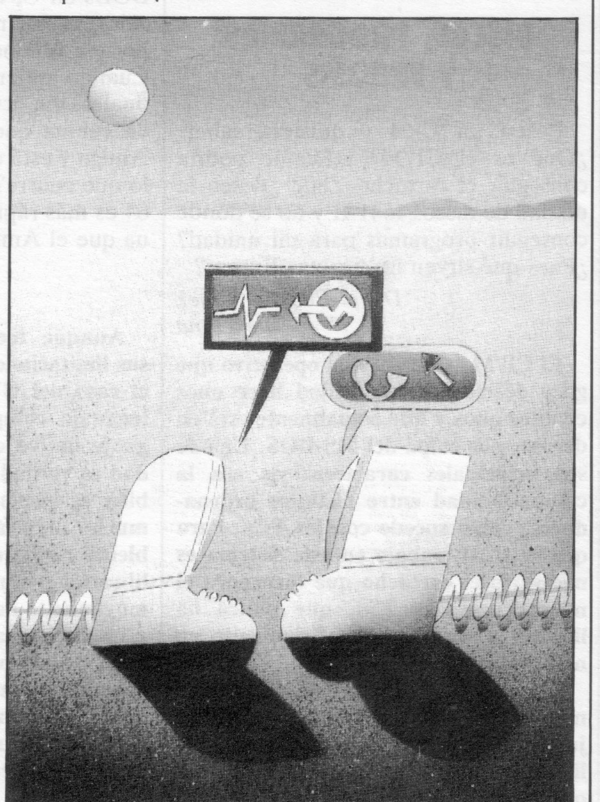
Además de la emulación/transformación a PC por hardware de productos como el Sidecar o la tarjeta para el Amiga 2000, y del «aprendiz» de emulador llamado Transformer, existe desde hace tiempo un curioso utilitario para la conversión de ficheros entre el Amiga y el PC.

DOS-2-DOS (que se lee DOS-two-DOS, y se pronuncia como DOS-to-DOS, «de DOS a DOS» en inglés) permite la lectura y grabación de discos en formato PC, el formato empleado por todos los ordenadores tipo IBM PC y compatibles. Las unidades de discos de 3 1/2″ se están imponiendo actualmente en el mercado, por su rapidez, mayor capacidad de almacenamiento y fiabilidad. Los IBM PS/2 ya vienen provistos de ellas y cualquier compatible admite unidades adicionales internas o externas en este pequeño formato.

Tras arrancar el programa, que se ejecuta desde el CLI, el prompt del sistema operativo del Amiga cambia a «D2D>». La única pregunta que hace el programa al comenzar es el nombre de la unidad de discos del Amiga que se va a emplear en modo PC. Los usuarios de varias unidades (DF1:, DF2:, etc.) pueden emplear cualquiera de ellas, dejando las demás en modo Amiga. Los que sólo dispongan de una unidad, por ejemplo la DF0: del Amiga 500, habrán de emplearla obligatoriamente en modo PC.

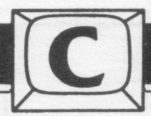
El aspecto externo de DOS-2-DOS es exactamente el mismo que el de una ventana CLI del AmigaDOS. Los comandos se introducen del mismo modo y utilizando el incómodo editor de líneas del CLI.

La transferencia de datos entre Amiga-PC puede hacerse copiando ficheros de un disco a otro. Es decir, que para pasar todos los ficheros de un disco situado en DF0: a un disco de PC en DF1:, el comando que habría que teclear es: «COPY DF0:#? DF1:». En el caso de los usuarios que tienen una sola unidad esto no es posible, y la única solución consiste en emplear el disco RAM: como «intermediario». Este sistema es recomendable (siempre que la memoria disponible sea suficiente) puesto que además agiliza el proceso. Por supuesto, la operación puede realizarse en ambos sentidos: de Amiga a PC o de PC a Amiga.



Existe una serie de comandos adicionales además del que permite copiar ficheros. Estos comandos, que funcionan tanto para los discos de Amiga como para los de PC, son los siguientes:

DIR
Muestra el directorio del disco.
CHDIR
Cambia de directorio en el disco.
TYPE
Muestra por pantalla un fichero de texto (ASCII).
COPY
Copia un fichero de un disco a otro, dentro del mismo disco.
DELETE
Borra un fichero.
FORMAT
Formatea discos en modo Amiga o PC.
HELP
Muestra un texto de ayuda con los comandos de DOS-2-DOS.
RESTART
Permite seleccionar nuevamente la unidad de discos que se utiliza en modo PC.



OMENTARIOS COMMODORE

EXIT
Salir de DOS-2-DOS.

Todos estos comandos permiten trabajar con los ficheros de múltiples formas, lo que facilita el mantenimiento perfecto de los discos. Si bien los comandos más comunes son COPY, DIR y FORMAT, los demás son también aprovechables. En algunos de ellos se incluye una conversión especial de formato ASCII, para pasar ficheros de texto sin problemas con los códigos empleados para caracteres especiales, como acentos, gráficos, etc. (como es bien sabido, el ASCII de Amiga no es estándar IBM como en los compatibles).

Las aplicaciones de DOS-2-DOS, pese a parecer «únicas» son varias. Para empezar, el traslado de ficheros entre un ordenador tipo Amiga y otro tipo PC puede realizarse con gran facilidad a través de discos floppy. El tiempo necesario para realizar la copia con DOS-2-DOS de un disco a otro es mucho menor que el que sería necesario para la transferencia de otro modo (por ejemplo a través de un cable serie RS-232). También hay usuarios que disponen de ambos ordenadores pero una sola impresora. Con DOS-2-DOS se puede dejar la impresora conectada en uno de ellos y, con un sencillo cambio de discos, imprimir los ficheros en cuestión. Esto es especialmente útil en el caso de, por ejemplo, los ficheros «de impresión» obtenidos con programas de diseño CAD-CAM o autoedición, listos para ser impresos en cualquier otro ordenador.

DOS-2-DOS permite copiar ficheros entre dos sistemas distintos como son el AmigaDOS y el MS-DOS.

Por si esto fuera poco, las versiones más modernas de DOS-2-DOS también admiten el mismo proceso de conversión y grabación a formato de discos ATARI, con lo que el campo abarcado por el Amiga se abre aún más.

En definitiva, un utilitario verdaderamente práctico al que no cabe duda que muchos usuarios encontrarán utilidad, sobre todo si tienen acceso a ambos sistemas, como es el caso de un gran número de apasionados de la informática. ■

DIRECTORIO

top 16

TU TIENDA AMIGA EN VALENCIA
CLUB DE USUARIOS
VENTA POR CORREO
SERVICIO 24 HORAS

Tel. (96) 326 40 90
San Francisco de Borja, 4, bajo. (Valencia)

CLIP INFORMATICA

DISTRIBUIDOR OFICIAL COMMODORE

DISPONEMOS DE TODA LA GAMA DE ORDENADORES Y PERIFERICOS COMMODORE. REALIZAMOS ANIMACIONES, TITULACIONES, MONTAJES EN VIDEO, ETC.

C/ Jenaro de la Fuente, 2
Tel. (986) 37 46 29
36205 VIGO

ELECTROAFICION

- Ordenadores de Gestión PC/XT/AT
- Commodore C-64, C-128, AMIGA
- Accesorios de Informática
- Software Gestión. Juegos
- Radioaficionados
- Comunicaciones
- Reparaciones COMMODORE

Villarroel, 104
08011 Barcelona - Tels.: 253 76 00-09

MADCOMPUTER

Los especialistas en AMIGA

Soluciones profesionales
- **Imagen:** Digitalización y tratamiento.
- **Video:** Ray-tracing, grafismo electrónico, animación. Genlocks.
- **Sonido y música:** Digitalización, composición y grabación/interpretación, MIDI.
- **Autoedición:** Impresión/Filmación láser PostScript.

Nicaragua, 4, bajo
Teléf. (91) 250 90 40. 28016 MADRID

TOT MICRO

C/. Forn St. Llucia, 1
08240 Manresa. Tel. (93) 872 22 97

DISTRIBUIDOR OFICIAL COMMODORE

AMIGA 500-2000 - PC. COMPATIBLES
DISCOS DUROS - AMPLIACIONES DE MEMORIA
DIGITALIZADORES - VIDEO-SONIDO
PLOTERS - IMPRESORAS
DISKETS 3 1/2-5 1/4, ETC.

HEROS INFORMATICA

AMIGA 500 Y 2000
SOFTWARE Y HADWARE
PERIFERICOS

INDEPENDENCIA, 350, 2.°
(93) 348 10 27 08026 BARCELONA

TEX-HARD, S.A.

AMIGA 500 Y 2000
SOFTWARE AMIGA
PC'S COMMODORE
IMPRESORAS
ACCESORIOS
PERIFERICOS

C/ Corazón de María, 9
Tels.: 416 95 62 - 416 96 12. 28002 Madrid.

NHS

FABRICAMOS EN ESPAÑA
BUSCAMOS DISTRIBUIDORES

- **DIGITALIZADOR AUDIO MONO**
- **DIGITALIZADOR AUDIO STEREO CON ENTRADA MICROFONO**
- **GENLOCK... etc.**

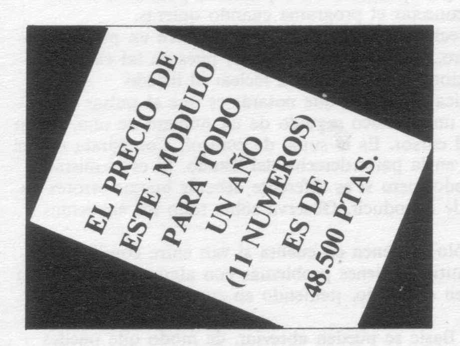
C/ Santa Anna, 11-13, 2.°, 2.° A
08002 BARCELONA
Tel. (93) 317 34 37. Fax (93) 318 50 83

DEFOREST microinformática

DISTRIBUIDOR OFICIAL COMMODORE

DISPONEMOS DE TODA LA GAMA DE ORDENADORES, IMPRESORAS Y PERIFERICOS COMMODORE. DISPONEMOS DE SOFT EN GENERAL.

BARCELONA
C/Viladomat, 105. Tel. 423 72 29



TELESOFT

CLUB DE USUARIOS DE AMIGA.
SOLICITA INFORMACION AL APTO. 658 ORENSE.
TE HAREMOS LLEGAR NUESTRO BOLETIN GRATUITAMENTE.

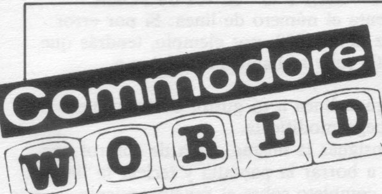


DISTRIBUIDOR OFICIAL COMMODORE

- **VENTA DE ORDENADORES**
- **CURSOS DE INFORMATICA**
- **CLUB DE USUARIOS AMIGA 500**
- **SERVICIO TECNICO**

ESPECIALISTAS EN REPARACION DE ORDENADORES COMMODORE CON 10 AÑOS DE EXPERIENCIA

NUEVA DIRECCION: C/ LIÑAN, 1
TEL. (976) 29 29 29, 50001 ZARAGOZA



¿QUIERES COLABORAR CON NOSOTROS?

Si quieres colaborar con nosotros, envíanos tus artículos a la siguiete dirección:
COMMODORE WORLD Colaboraciones.
Rafael Calvo, 18-4.° B. 28010 MADRID.

CLAVE PARA INTERPRETAR LOS LISTADOS

Todos los listados que se publican en Commodore World han sido cuidadosamente comprobados en el modelo correspondiente de los ordenadores Commodore. Para facilitar su edición y para mejorar la legibilidad por parte del usuario se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Dado que los ordenadores Commodore utilizan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos, así como movimientos del cursor, códigos de color, etc., por equivalencias entre corchetes que indican la secuencia de teclas que se debe pulsar para obtener dichos caracteres.

Las claves constan siempre de una letra o símbolo precedidos de las palabras COMM, SHIFT o CTRL; por ejemplo [COMM +] o [SHIFTA]. Esto indica que para obtener el gráfico hay que pulsar a la vez la tecla COMMODORE (la de abajo a la izquierda) o la tecla SHIFT (cualquiera de ellas) junto con la letra o símbolo correspondiente, en este ejemplo "+" o "A".

También puede aparecer un número indicando cuántas veces hay que repetir el carácter.

[7 CRSRR] equivale a siete cursores a la derecha y [3 SPC] a tres pulsaciones de la barra espaciadora.

```
1 REM "PERFECTO"                      .113
2 REM POR JAMES E. BORDEN, RUN EEUU .96
3 REM (C)1986 COMMODORE WORLD        .157
4 :                                   .236
5 POKE56,PEEK(56)-1:POKE52,PEEK(56) .119
6 CLR:PG=PEEK(56):ML=PG*256+60      .232
7 :                                   .239
8 P=ML:L=24                           .216
9 S=0:FORI=0TO6:READA:IFA=-1THEN16  .59
10 IFA<0ORA>255THEN14                 .146
11 POKEP+I,A:S=S+A:NEXT              .81
12 READSC:IFS<>SCTHEN14              .250
13 L=L+1:P=P+7:GOTO9                 .97
14 PRINT"ERROR EN DATAS LINEA";L:EN .60
D
15 :                                  .247
16 POKEML+4,PG:POKEML+10,PG          .60
17 POKEML+16,PG:POKEML+20,PG         .221
18 POKEML+32,PG:POKEML+38,PG         .110
19 POKEML+141,PG                      .97
20 SYSML:PRINT"[CRSRD] [WHT]CORRECT .98
OR ACTIVADO
21 PRINT" SYS"ML"=CONECTAR            .127
22 PRINT" SYS"ML+30"=DESCONECTAR[CO .122
MM7]
23 :                                  .255
24 DATA173,5,3,201,3,208,1,594        .22
25 DATA96,141,105,3,173,4,3,525       .181
26 DATA141,104,3,162,103,160,3,676    .214
27 DATA142,4,3,140,5,3,96,393         .177
28 DATA234,234,173,104,3,141,4,893    .96
29 DATA3,173,105,3,141,5,3,433        .177
30 DATA96,32,124,165,132,11,162,722 .18
31 DATA0,142,240,3,142,241,3,771      .87
32 DATA189,0,2,240,51,201,32,715      .166
33 DATA208,4,164,212,240,40,201,106 .177
9
34 DATA34,208,8,72,165,212,73,772     .146
35 DATA1,133,212,104,72,238,241,100 .237
1
36 DATA3,173,241,3,41,7,168,636       .142
37 DATA104,24,72,24,104,16,1,345      .225
38 DATA56,42,136,16,246,109,240,845 .238
39 DATA3,141,240,3,232,208,200,1027 .123
40 DATA173,240,3,24,101,20,24,585     .72
41 DATA101,21,141,240,3,169,42,717    .49
42 DATA32,210,255,169,0,174,240,108 .170
0
43 DATA3,32,205,189,162,4,189,784     .83
44 DATA211,3,32,210,255,202,16,929    .214
45 DATA247,164,11,96,145,13,32,708    .87
46 DATA32,0,0,0,0,0,32,-1             .146
```

CLAVE	EQUIVALENCIA
CRSRD	CURSOR ABAJO (SIN SHIFT)
CRSRU	CURSOR ARRIBA (CON SHIFT)
CRSRR	CURSOR DERECHA (SIN SHIFT)
CRSRL	CURSOR IZQUIERDA (CON SHIFT)
HOME	CLR/HOME SIN SHIFT
CLR	CLR/HOME CON SHIFT
SPC	BARRA ESPACIADORA
DEL	INST/DEL Y SHIFT + INST/DEL
INST	INST/DEL CON SHIFT
BLK A YEL	COLORES: CONTROL + NUMERO
RVS ON	CONTROL + 9
RVS OFF	CONTROL + 0
F1 A F8	TECLAS DE FUNCION
FLCH ARRIBA	FLECHA ARRIBA
FLCH IZQ	FLECHA A LA IZQUIERDA
PI	PI (FLECHA ARRIBA CON SHIFT)
LIBRA	LIBRA
PARA C-128	
BELL	CONTROL + G
TAB	TAB O CONTROL + I
LFEED	LINE FEED O CONTROL + J

Cómo utilizar la suma de control

Todos los listados para C-64 o C-128 que aparecen en la revista llevan una suma de control para que no te equivoques a la hora de teclearlos. Para poder utilizar esta suma de control tendrás que teclear el programa que aparece aquí listado. Se llama "Perfecto". Cuando lo hayas tecleado, compruébalo y sálvalo en disco o cinta.

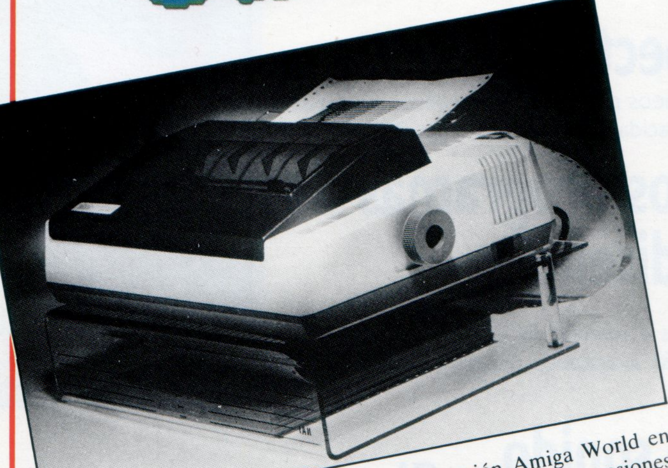
Cada vez que vayas a teclear uno de los programas que aparecen en Commodore World debes cargar antes el programa corrector. Apunta los dos valores SYS que aparezcan en la pantalla, pues te servirán para conectar o desconectar el programa cuando quieras.

Verás que a la derecha de todos los listados aparece un punto seguido de un número. Eso no debes teclearlo, pues en tal caso el programa no te funcionaría. Comienza a teclear el listado normalmente. La única diferencia que notarás es que al pulsar RETURN aparecerá un asterisco seguido de un número de una, dos o tres cifras debajo del cursor. Es la suma de control. Compárala con el número que aparece en la parte derecha del listado. Si es el mismo, puedes seguir tecleando, pero si es diferente deberás buscar errores en la línea que acabes de introducir. Observa sobre todo los siguientes puntos:

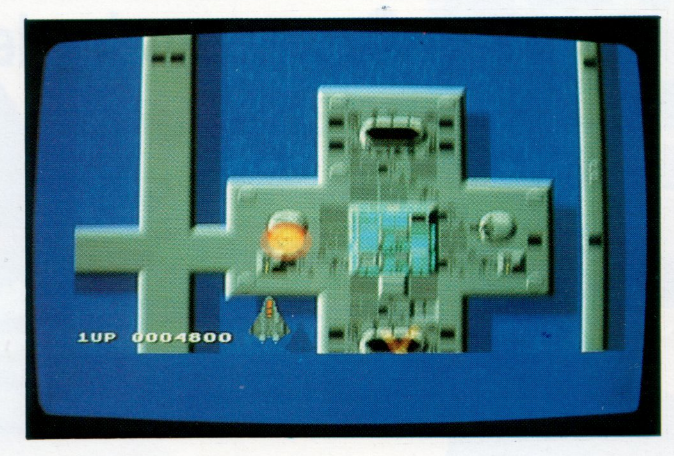
- Los espacios sólo se tienen en cuenta si van entre comillas. Los demás los puedes omitir. Si tienes problemas con alguna línea tecléala tal y como aparece en el listado, ¡teniendo en cuenta las claves, por supuesto!
- Los comandos Basic se pueden abreviar, de modo que puedes poner ? en vez de PRINT o P SHIFT O en vez de POKE.
- También se tiene en cuenta el número de línea. Si por error introduces la línea 100 en vez de la 1000, por ejemplo, tendrás que teclear nuevamente la línea 100 (que se habrá borrado) y a continuación la 1000.
- Si quieres modificar alguna línea a tu gusto, obtén primero la suma de control correcta y luego modifícala.

Si por alguna razón no consigues la misma suma de control que aparece en el listado, prueba a borrar la pantalla y teclear la línea entera de nuevo. Un artículo completo sobre el funcionamiento de este programa apareció en el número 23 de Commodore World.









Ya está a la venta el cuarto disco de la sección Amiga World en nuestra revista Commodore World. El disco, como en ocasiones anteriores va acompañado por otro de regalo, además de incluir un estupendo programa generador de caracteres definidos por el usuario, para modificar el juego de caracteres NLQ de su impresora. Todos los listados de Amiga aparecidos en los números 55, 56 y 57 de Commodore World, podrás encontrarlos en el disco Amiga 4. ¡¡¡ATENCION!!! El disco de regalo de este mes contiene un fenomenal juego comercial **Sidewinder.** En realidad se trata de una parte del total de este juego, pero que permite jugar en los dos primeros niveles. Una demo **jugable.** Cedido por DRO SOFT.

POR SOLO 1.700 PESETAS DOS DISCOS CON UN MONTON DE PROGRAMAS Y LISTADOS, Y ADEMAS UN JUEGO COMERCIAL DE REGALO.

BOLETIN DE PEDIDO - DISCOS AMIGAWORLD IV

Nombre ....

Dirección ....

Población .... C.P. .... Provincia ....

Teléfono .... Modelo de Amiga ....

☐ Deseo recibir el disco AMIGA World 4 (1.700 ptas.)

☐ Deseo suscribirme por un año (4 discos y 11 revistas) al precio especial de 9.000 ptas., a partir de la revista número ....

☐ Incluyo cheque por .... ptas.

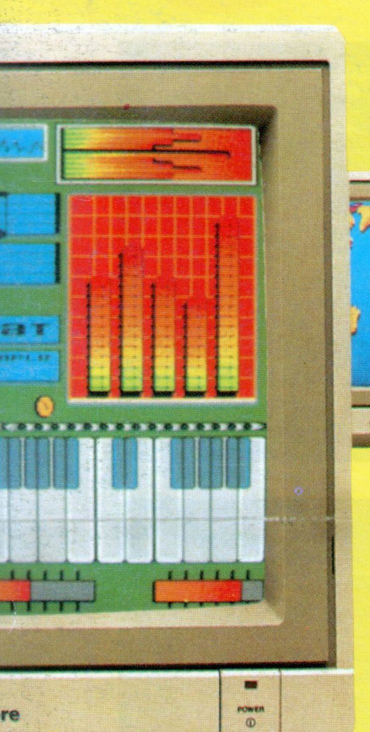
☐ Incluyo giro número .... por .... ptas.

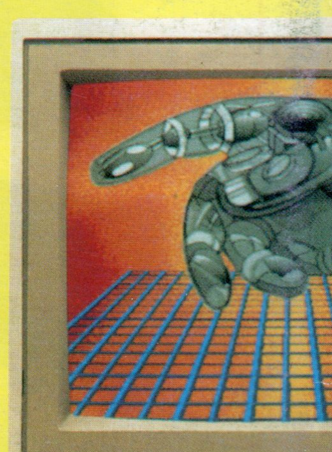
Enviar a: COMMODORE WORLD, Rafael Calvo, 18, 4.º B. 28010 Madrid.

Forma de pago: Sólo cheque giro. No se sirven pedidos contrarreembolso. Gastos de envío incluidos.

La suscripción se puede pagar por tarjeta VISA o MASTERCARD, bien por carta o por teléfono.

EL COMMODORE AMIGA 500 demuestra fácilmente que es el líder de los ordenadores domésticos





4.096 colores espectaculares

4.096 colores simultáneos proporcionan gráficos de calidad profesional y con capacidad de animación en 3 D.

Fantásticos programas de dibujo y diseño

La gran capacidad de gráficos y manejo de colores permiten realizar los más avanzados programas de diseño y dibujo.

Cuatro canales de sonido estéreo

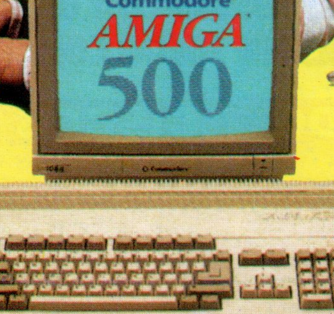
Cuatro canales; sonido digitalizado, sonido estéreo y sintetizador de voz incorporado, proporcionan una serie de capacidades de sonido que sólo se pueden encontrar en Amiga.

Sistema multiárea

El COMMODORE AMIGA es el único ordenador personal que permite rodar simultáneamente varios programas.











Calidad en videojuegos

Calidad exclusiva en gráficos de videojuegos y sonido estéreo, hacen del AMIGA 500 un centro de entretenimiento y diversión.

CERTIFICADO DE DESCUENTO

800 PESETAS

AHORRO DE 800 PTAS. AL HACERSE SUSCRIPTOR A COMMODORE WORLD (18 % Dto.)

¡NO DESAPROVECHE ESTA MAGNIFICA OCASION!

TARJETA DE PEDIDO

Nueva suscripción ☐ Renovación ☐

Nombre ....

Empresa ....

Dirección ....

Población .... C.P. ....

Provincia ....

Teléf. .... Modelo ordenador ....

Deseo iniciar mi suscripción con el N.º ....

Marque con una (X) la forma de pago elegida.

☐ Adjunto cheque de 3.600 ptas.

☐ Adjunto giro n.º .... por 3.600 ptas.

TARJETA VISA ☐ MASTERCARD ☐

N.º tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
(No olvide incluir los 16 números)

Fecha caducidad ....

Firma

Una suscripción a COMMODORE WORLD durante un año por sólo 3.600 ptas. (11 ejemplares)

Recorte y envíenos por correo esta tarjeta después de haberla rellenado correctamente

800 ptas.

Descuento

800 ptas.

Descuento

800 ptas.

Descuento

800 ptas.

Descuento

UN AÑO RECIBIENDO COMMODORE WORLD

UD. SOLO PAGA 3.600 PTAS.

RESPUESTA COMERCIAL
Autorización n.º 6.083
B. O. de C. y T. de
11-4-86

C W COMMUNICATIONS, S.A.
APARTADO 353 F.D.
28080 MADRID

NO NECESITA SELLO
A franquear en destino

CERTIFICADO DE DESCUENTO

800 PESETAS

AHORRO DE 800 PTAS. AL HACERSE SUSCRIPTOR A COMMODORE WORLD (18 % Dto.)

¡NO DESAPROVECHE ESTA MAGNIFICA OCASION!

TARJETA DE PEDIDO

Nueva suscripción ☐ Renovación ☐

Nombre ....

Empresa ....

Dirección ....

Población .... C.P. ....

Provincia ....

Teléf. .... Modelo ordenador ....

Deseo iniciar mi suscripción con el N.º ....

Marque con una (X) la forma de pago elegida.

☐ Adjunto cheque de 3.600 ptas.

☐ Adjunto giro n.º .... por 3.600 ptas.

TARJETA VISA ☐ MASTERCARD ☐

N.º tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

(No olvide incluir los 16 números)

Fecha caducidad ....

Firma

Una suscripción a COMMODORE WORLD durante un año por sólo 3.600 ptas. (11 ejemplares)

Recorte y envíenos por correo esta tarjeta después de haberla rellenado correctamente

UN AÑO RECIBIENDO COMMODORE WORLD

UD. SOLO PAGA 3.600 PTAS.

